

SCO[®] Open Desktop[®] /
SCO Open Server[™]
System Administrator's Guide
Operating System,
Networking, and DOS Services
Includes Performance and Troubleshooting

SCO® OPEN DESKTOP®/SCO® OPEN SERVER™ Software

© 1983-1992 The Santa Cruz Operation, Inc. All Rights Reserved

The copyrighted software that accompanies this manual is licensed to the End User only for use in strict accordance with the End User License Agreement, which should be read carefully before commencing use of the software.

SCO OPEN DESKTOP/SCO OPEN SERVER Software is commercial computer software and, together with any related documentation, is subject to the restrictions on U.S. Government use as set forth below.

If this procurement is for a DOD agency, the following DFAR Restricted Rights Legend applies:

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause at DFARS 52.227-7013. "Contractor/ manufacturer" is The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, California, 95060, U.S.A.

If this procurement is for a civilian government agency, the following FAR Restricted Rights Legend applies:

RESTRICTED RIGHTS LEGEND: This computer software is submitted with restricted rights under Government Contract No. _____ (and Subcontract No. _____, if appropriate). It may not be used, reproduced, or disclosed by the Government except as provided in Paragraph (g)(3)(i) of FAR Clause 52.227-14 or as otherwise expressly stated in the contract. Contractor/Manufacturer is The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, CA 95060.

SCO OPEN DESKTOP/SCO OPEN SERVER contains software licensed from a number of sources. The following are copyright notices for the software from these contributions which is used in SCO OPEN DESKTOP/SCO OPEN SERVER.

SCO OPEN DESKTOP/SCO OPEN SERVER Operating System Software: © 1983-1992 The Santa Cruz Operation, Inc.; © 1981-1989 Microsoft Corporation; © 1978-1989 UNIX System Laboratories, Inc.; © 1988-1989 Secureware Inc.; © 1989 Acer Corporation. All Rights Reserved.

SCO OPEN DESKTOP/SCO OPEN SERVER Networking and Communication Software: © 1984-1989 Microsoft Corporation; © 1987-1988 Lachman Technology, Inc.; © 1987 Convergent Technologies Inc.; © 1986 Sun Microsystems Inc.; © 1986-1992 The Santa Cruz Operation, Inc. All Rights Reserved.

SCO OPEN DESKTOP/SCO OPEN SERVER Windowing and Graphic User Interface Software: © 1988-1989 Locus Computing Corporation; © 1985-1989 Metagraphics Software Corporation; © 1989 Open Software Foundation, Inc.; © 1988-1989 IXI Limited; © 1989 Sun Microsystems, Inc. © 1985-86 Bigelow & Holmes. © 1988-1992 The Santa Cruz Operation, Inc. All Rights Reserved.

SCO OPEN DESKTOP/SCO OPEN SERVER MS-DOS Integration Software: © 1982-1989 Microsoft Corporation; © 1985-1989 Locus Computing Corporation; © 1989-1992 The Santa Cruz Operation, Inc. All Rights Reserved.

SCO NFS was developed by Lachman Technology, Inc. based on LACHMAN SYSTEM V NFS. LACHMAN is a trademark of Lachman Technology, Inc. NFS is a trademark of Sun Microsystems, Inc.

SCO TCP/IP was developed by Lachman Technology, Inc. SCO TCP/IP is derived from LACHMAN SYSTEM V STREAMS TCP, a joint development of Lachman Technology, Inc. and Convergent Technologies. LACHMAN is a trademark of Lachman Technology, Inc.

SCO OPEN DESKTOP/SCO OPEN SERVER Administration and User Documentation: © 1983-1992 The Santa Cruz Operation, Inc.; © 1980-1989 Microsoft Corporation; © 1988 UNIX Systems Laboratories, Inc.; © 1985-1989 Locus Computing Corporation; © 1987-1989 Lachman Technology, Inc.; © 1987 Convergent Technologies, Inc.; © 1981-1989 ASK Computer Systems, Inc.; © 1989 Open Software Foundation, Inc.; © 1989 Digital Equipment Corporation, Maynard, Mass.; © 1987-1989 Hewlett-Packard Company; © 1988 Massachusetts Institute of Technology; © 1985-1989 Metagraphics Software Corporation. All Rights Reserved.

No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of the copyright owner, The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, California, 95060, U.S.A. Copyright infringement is a serious matter under the United States and foreign Copyright Laws.

Information in this document is subject to change without notice and does not represent a commitment on the part of The Santa Cruz Operation, Inc.

SCO OPEN DESKTOP/SCO OPEN SERVER Software is commercial computer software and, together with any related documentation, is subject to the restrictions on U.S. Government use as set forth below.

If this procurement is for a DOD agency, the following DFAR Restricted Rights Legend applies.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause at DFARS 52.227-7013. "Contractor/ manufacturer" is The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, California, 95060, U.S.A.

If this procurement is for a civilian government agency, the following FAR Restricted Rights Legend applies:

RESTRICTED RIGHTS LEGEND: This computer software is submitted with restricted rights under Government Contract No. _____ (and Subcontract No. _____, if appropriate). It may not be used, reproduced, or disclosed by the Government except as provided in Paragraph (g)(3)(i) of FAR Clause 52.227-14 or as otherwise expressly stated in the contract. Contractor/Manufacturer is The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, CA 95060.

Open Desktop, the Open Desktop logo, SCO, the SCO logo, and The Santa Cruz Operation are registered trademarks of The Santa Cruz Operation, Inc. in the U.S.A. and other countries. Open Server is a trademark of The Santa Cruz Operation, Inc.

UNIX and OPEN LOOK are registered trademarks of UNIX System Laboratories, Inc. in the U.S.A. and other countries.

INGRES and INGRES/386 are trademarks of ASK Computer Systems, Inc.

MS-DOS, XENIX and Microsoft are registered trademarks of Microsoft Corporation in the U.S.A. and other countries and Flight Simulator is a trademark of Microsoft Corporation.

NFS is a trademark of Sun Microsystems, Inc.

OSF is a trademark of The Open Software Foundation, Inc.

Xsight is a registered trademark and Merge 386 is a trademark of Locus Computing Corporation.

Lucida is a registered trademark of Bigelow and Holmes.

Product names mentioned which are not the property of The Santa Cruz Operation, Inc. are/may be trademarks or registered trademarks of their respective companies.

Date: 16 April 1993

Document version: 3.0.0A

About this guide

SCO® Open Desktop®/SCO Open Server™ has three administrator's guides.

This *System Administrator's Guide* covers the administration of:

- Operating System Services
- DOS Services
- Networking Services
- Performance Management and Troubleshooting

The other Open Desktop administrator's guides are:

- *Graphical Environment Administrator's Guide*

This book discusses customizing and administering the graphical portion of SCO Open Desktop/SCO Open Server.

- *Open Server System Administrator's Guide*

This manual is optional for SCO Open Desktop. It contains information on administering SCO Open Server (including PC-Interface and IPX/SPX), plus additional documentation on system security, backups, UNIX system initialization files, and a DOS *terminfo* reference.

Administering Operating System Services

Chapter 1

Introduction 9

The system administrator and administrative roles	9
Making administration easier with the sysadmsh	11
The super user account	12
The keyboard	12
Running programs simultaneously with MultiScreen	13
System security	14
Site planning considerations	14

Chapter 2

Using the system administration shell 17

Starting sysadmsh	17
How the screen is organized	19
Selecting menu items	20
Using forms	22
Using radio buttons	28
Using scan windows	29
Getting help	31
Changing the current directory within sysadmsh	33
The function keys	33
Using shell escapes to access the UNIX system command line	34
Customizing sysadmsh menus	34
sysadmsh menu options	35

Chapter 3

Starting and stopping the system 43

Starting the system	43
Logging in as the super user	51
Stopping the system	52
Understanding the boot display information	53
Changing the boot process	54

Chapter 4

Administering user accounts **57**

How account information is stored	58
Account management	60
Default account configuration	81

Chapter 5

Managing filesystems **95**

What is a filesystem?	95
Mounting and unmounting a filesystem	97
Permitting users to mount filesystems	98
Configurable filesystem features	99
Maintaining free space in filesystems	100
Maintaining efficient filesystem organization	108
Adding disk space and filesystems	111
Filesystem integrity	111

Chapter 6

Using printers **119**

The printerspooling system	119
Printer configuration options	120
Print service command summary	134
Starting and stopping the print service	135
Canceling a print request	137
Enabling and disabling printers	137
Adding a printer to a class	138
Setting the system default destination	139
Mounting a form or print wheel	139
Removing a printer or class	141
Managing the printing load	141
Managing queue priorities	143
Using forms and filters	148
Using the information in the request log	149
Customizing the print service	150
Configuring a network printer	162

Configuring a dialup printer	163
Using a printer without the spooler	165
Creating an init device file	165

Chapter 7

Using floppy disks and tape drives **167**

Using tape drives	167
Using floppy disks	173

Chapter 8

Using MS-DOS and other DOS operating systems **179**

OS/2 coexistence	180
Partitioning the hard disk using fdisk	180
Installing a UNIX system partition on a DOS system	184
Using a UNIX system and DOS with two hard disks	185
Removing an operating system from the hard disk	185
DOS accessing utilities	186
Mounting DOS filesystems on a UNIX system	187

Administering DOS Services

Chapter 1

Introduction **203**

Who should use this guide	203
Organization of this guide	204

Chapter 2

Administering DOS Services **205**

Using the dosadmin program	206
Adding and deleting user accounts	206

Administering the system console	206
Administering terminals	210
Administering COM ports	212
Administering DOS printers	212
Backing up filesystems	217
Administering disk and diskette drives	217
Administering physical DOS partitions	219
Administering virtual DOS partitions and virtual floppy disks	223
Installing plug-in cards in your computer	228
Making new DOS images	233
System files affected by system administration	238

Chapter 3

Installing DOS applications 239

Tips on installing DOS applications to run under DOS Services	240
Installing applications with advanced processor requirements	243
Configuring DOS applications for use from the UNIX shell	245
Installing copy-protected DOS applications	248
Removing DOS applications	252

Chapter 4

Tuning applications with the merge command 253

Chapter 5

Tailoring the operation of DOS 261

DOS defaults	262
Using DOS options on the dos command line	262
Installing DOS options to change defaults	263
Installing DOS options to override system defaults	264
Using dosadmin to install and remove DOS options	265
Using dosopt to install and remove DOS options	271
Allocating memory for DOS	276
Allocating memory for windows	278

Chapter 6***DOS options*****281**

Attach devices to DOS	284
Display-oriented and stream-oriented DOS programs	291
Pass command directly to DOS command interpreter	293
Set initial current drive	294
Interpret configuration file	294
Help text	295
Alternative DOS load image	296
DOS memory size	297
Run autoexec.bat	298
Change DOS printer timeout	299
Translate DOS switch characters and path separators	300
Dosopt "verbose" mode	301
Set X Client startup	302
dosopt null option	302
Remove assigned options	302

Chapter 7***Using DOS Services NLS features*****305**

DOS code pages and UNIX operating system code sets	308
How MS-DOS handles NLS	309
How the UNIX operating system handles NLS	310
How DOS Services handles NLS	311
Sample NLS setup procedures for VGA displays	316
Sample NLS setup procedures for ISO terminals	319
Naming DOS and UNIX operating system files	320
Printing in an NLS environment	321
Converting text files	322

Chapter 8***Other advanced topics*****325**

Changing colors in DOS windows	326
X System resources for DOS Services	328
Using expanded memory (EMS)	331

Using multiport serial cards	331
DOS Services/DOS compatibility	332
Passing UNIX environment variables to DOS	335
Running DOS or a DOS application as the default shell	339
Scheduling DOS programs	340

Appendix A

Using the system console and terminals 341

Characteristics of DOS on the console	341
Using EGA cards	342
Using VGA cards	344
Using PC scancode terminals	344
Using serial terminals	345
Console and terminal control codes	349

Appendix B

Terminfo guide 351

Basic concepts	352
Output on standard terminals	355
Input on standard terminals	362
About PC scancode terminals	366
Output on PC scancode terminals	370
Input on PC scancode terminals	373
PC-character-to-ASCII conversions	375
terminfo escape sequences	376

Administering Networking Services

Chapter 1

Networking overview 401

Introduction to networking	402
How the network works	403
Networking services	407
Introducing TCP/IP	408

Introducing NFS	418
Introducing LAN Manager client	419
Introducing UUCP	421
Introducing MMDF	422
Other networking software	422

*Chapter 2****TCP/IP network administration*** **423**

Kernel configuration	423
Setting interface parameters	427
Creating a subnetwork	427
Network servers	428
Network databases	429
Administering pseudo ttys	432
Network tuning and troubleshooting	433

*Chapter 3****TCP/IP serial line communications*** **441**

Administering SLIP	441
Administering PPP	450

*Chapter 4****Configuring the BIND name server*** **461**

The name service	461
Types of servers	462
Setting up your own domain	464
Remote servers	466
Initializing the cache	466
Standard files	467
Standard resource records	467
Some sample files	473
Additional sample files	477
Domain management	479

Chapter 5
Gateways and routing **483**

Running routed	484
Running gated	485
More gated information	489

Chapter 6
Configuring and using SNMP **491**

Basic concepts	491
Other concepts	495
Overview of the SCO implementation	496
Configuring the SNMP agent	497
Using the SNMP commands	499
Using SNMP to correct problems	504
More SNMP information	506

Chapter 7
Remote line printing **507**

Installing and removing RLP	508
How RLP works	509
Using RLP	510
Setting up a client	511
Setting up a print server	514
Deleting printcap entries	515

Chapter 8
Synchronizing clocks **517**

Time synchronization protocol	517
Network time protocol	521

*Chapter 9****Administering NFS* 547**

Basic concepts	547
Configuring NFS	549
Administering NFS	554

*Chapter 10****NIS Network Information Service* 565**

Understanding NIS	565
Planning an NIS configuration	571
Configuring NIS	573
Managing NIS maps	582
Managing user accounts	588
Administering an NIS network	592
Directory of NIS services	603

*Chapter 11****Automounter* 607**

Understanding Automounter	607
Configuring Automounter	613
Administering Automounter	627

*Chapter 12****LAN Manager client administration* 635**

Understanding NetBIOS	635
Adding and removing a client	640
Configuring the client for user access to servers	640
Starting and stopping the client	641
Managing the client	642
Utility administration on the client	645
Modifying client parameters	646

Chapter 13

Building a remote network with UUCP

659

What is UUCP?	659
How to use this chapter	660
What you need	661
UUCP commands	661
Configuring UUCP on your system	665
Detailed descriptions of UUCP configuration files	669
Special UUCP configuration options	687
Complete UUCP examples	697
Administering your UUCP system	701
UUCP error messages	704

Chapter 14

Setting up electronic mail

711

Chapter overview	712
How MMDF works	712
The configuration utility	717
MMDF configuration checklist	719
Running the configuration utility	727
MMDF configuration files	739
How MMDF routes mail	755
Modifying MMDF configuration	759
Testing MMDF configuration	769
Maintaining the MMDF system	770
Advanced MMDF configuration tasks	771
Getting more information	786

Chapter 15

sendmail administration

789

sendmail and other mailers	790
How sendmail works	792
Configuring sendmail	795
Running sendmail	810
Administering sendmail	817
More sendmail information	823

*Chapter 16****Using modems*** **825**

Choosing a serial port	825
Configuring your modem	826
Installing a modem	828
Dialing into your computer	834
Dialing out from your computer	835

*Chapter 17****Helpful Hints*** **843**

Setting the broadcast address	843
Problem with WD8003 card	844
Making remote backups	844
Differences in sendmail implementations	846
Setting up user equivalence	847

Performance and Troubleshooting***Preface*** **857**

Audience	857
How to use this book	857

*Chapter 1****Troubleshooting overview*** **859**

Avoiding system problems	859
Preparing to recover from problems	861
Kernel error messages	861
Recovering from a system crash	863

Chapter 2

Performance management 867

General performance considerations	867
Understanding your performance goals	869
Excessive paging	870
Kernel tables are too small	871
Tuning disk I/O parameters	872
Optimizing serial I/O	875
Reducing memory dedicated to graphical clients	880

Chapter 3

Tools reference 881

configure(ADM) — reallocate kernel resources	882
crash(ADM) — view memory	883
fsck(ADM) — check and repair filesystems	886
idtune(ADM) — reallocate kernel resources	887
integrity(ADM) — check system file permissions	887
ps(C) — check process activity	888
sar(ADM) — system activity reporter	889
swap(ADM) — check and add swap space	901
timex(ADM) — examine system activity per command	901
tunesh(ADM) — automatic configuration tool	902
XENIX system tools	903

Chapter 4

Kernel parameter reference 905

Disks and buffers	908
Character buffers	911
Files, inodes, and filesystems	912
Processes, memory management, and swapping	915
Clock	918
MultiScreens	918
Message queues	919
Semaphores	920
Shared data	921

System name	922
STREAMS data	922
Event queues and devices	925
Hardware dependent parameters	926
Security	927
Asynchronous I/O	928
Boot load extension parameters	929
Multiprocessing parameters	929
Network-specific parameters	930

Chapter 5

Troubleshooting reference 931

Boot troubleshooting	931
Console troubleshooting	939
cron, at, and batch troubleshooting	941
Disk problems	942
Filesystem troubleshooting	944
Login problems	955
Mail troubleshooting	957
Modem problems	959
Modem — errors when dialing out	959
Modem — problems dialing in	962
Mouse (serial) troubleshooting	965
Mouse (X server) troubleshooting	966
No more processes error message	966
Print system troubleshooting	967
Runaway processes, stopping	974
Security system troubleshooting	975
Tape drive problems	979
Terminal troubleshooting	982
UUCP troubleshooting	985

Index	997

Administering Operating System Services

Chapter 1***Introduction*****9**

The system administrator and administrative roles	9
Making administration easier with the sysadmsh	11
The super user account	12
The keyboard	12
Running programs simultaneously with MultiScreen	13
System security	14
Site planning considerations	14

Chapter 2***Using the system administration shell*****17**

Starting sysadmsh	17
How the screen is organized	19
Selecting menu items	20
Using forms	22
Using radio buttons	28
Using scan windows	29
Getting help	31
Changing the current directory within sysadmsh	33
The function keys	33
Using shell escapes to access the UNIX system command line	34
Customizing sysadmsh menus	34
sysadmsh menu options	35

Chapter 3***Starting and stopping the system*****43**

Starting the system	43
Loading the operating system	44
Cleaning filesystems	44
Choosing the mode of system operation	45
Entering system maintenance mode by shutting down first	46
Entering system maintenance mode directly	46
Setting the time and date	46
Checking the security databases	47

Logging in as the super user	51
Stopping the system	52
Using the shutdown command	52
Understanding the boot display information	53
Changing the boot process	54
Booting an old kernel	54
Changing the default bootstring (DEFBOOTSR)	55
Booting automatically	55

Chapter 4

Administering user accounts

57

How account information is stored	58
Configuring database recovery and precedence	59
Account management	60
Adding a user	60
Modifying account creation defaults	64
Sharing home directories	66
Adding administrative users	66
Altering the defaults displayed for user accounts	67
Removing or retiring a user account	67
Unretiring a user account	67
Locking or unlocking a user account	68
Locking or unlocking a terminal	69
Changing a user's login group	69
Changing a home directory	70
Changing a user password or password parameters	71
Altering user password expiration parameters	72
Changing user authorizations	74
Changing user audit parameters	75
Adding or changing groups	76
Supplemental groups	77
Changing the maximum number of supplemental groups	77
Allowing users to execute super user commands	77
Accessing other accounts with su(C)	78
Moving user accounts to another filesystem or directory	78
Migrating user accounts to non-SCO-based UNIX systems	79
Password compatibility	79
Migrating user accounts to SCO-based UNIX systems	80
Migrating user accounts from SCO XENIX or non-SCO-based UNIX systems	80

Default account configuration	81
Changing the security defaults	83
Changing system account parameters dynamically	85
Changing default login restrictions	85
Changing default password restrictions	87
Customizing password checking with goodpw(ADM)	89
Changing default authorizations	90
Subsystem authorizations	91
Kernel authorizations	92
Kernel authorizations and administrative users	93
Adding login shells and configuration files	94

Chapter 5

Managing filesystems 95

What is a filesystem?	95
Adding new filesystem(s)	96
Mounting and unmounting a filesystem	97
Permitting users to mount filesystems	98
Configurable filesystem features	99
Setting directory SGID bit	99
Setting filename truncation	99
Maintaining free space in filesystems	100
Strategies for maintaining free space	100
Displaying free space	101
Sending a system-wide message	101
Displaying disk usage	102
Displaying blocks by owner	102
Locating files	103
Finding temporary files	104
Finding files of a certain size	104
Finding files by permissions	104
Executing commands based on find output	104
Locating core files	105
Checking and clearing log files	106
Automatic log cleaning	107
Maintaining efficient filesystem organization	108
Disk fragmentation	109
Large directories	109
Limiting login directory size	110

Removing excess inode numbers	110
Adding disk space and filesystems	111
Filesystem integrity	111
How UNIX systems maintain files	112
How UNIX systems maintain filesystems	113
Causes of filesystem corruption	114
Rules for checking filesystems	114
Repairing filesystems with fsck	115
Summary of fsck phases	116
Automatic filesystem check	117

Chapter 6

Using printers

119

The printer spooling system	119
Printer configuration options	120
Interface program	121
Printer type	122
Content types	123
Connection method	125
Character sets or print wheels	125
Alerting to mount a print wheel	127
Fault alerting	129
Fault recovery	131
Restricting user access to a printer	132
Default printing attributes	133
Print service command summary	134
Starting and stopping the print service	135
Manually stopping the print service	136
Manually starting the print service	136
Canceling a print request	137
Enabling and disabling printers	137
Adding a printer to a class	138
Setting the system default destination	139
Mounting a form or print wheel	139
Removing a printer or class	141
Managing the printing load	141
Rejecting requests for a printer or class	142
Accepting requests for a printer or class	142
Moving requests to another printer	143

Managing queue priorities	143
Setting priority limits	144
Setting a default priority	145
Examining the priority limits and defaults	145
Moving a request around in the queue	145
Changing the priority for a request	146
Putting a request on hold	146
Moving a request to the head of the queue	147
Examining a printer configuration	147
Using forms and filters	148
Using the information in the request log	149
Customizing the print service	150
Adjusting the printer port characteristics	152
Adjusting the terminfo database	154
How to write an interface program	156
What does an interface program do?	157
How an interface program is used	157
Customizing the interface program	159
How to add an interface program	162
Configuring a network printer	162
Configuring a dialup printer	163
Using a printer without the spooler	165
Creating an init device file	165

Chapter 7

Using floppy disks and tape drives 167

Using tape drives	167
Tape device files	167
Changing the default tape drive	168
Setting the default tar(C) device	169
QIC cartridge drives	169
Mini-cartridge (Irwin) drives	170
DAT and 8-mm (Exabyte) drives	170
Archiving files on tape	170
The tar command	170
Tape drive maintenance	171
Irwin drive commands	172
Tape driver error correction code (ECC) support	172
QIC-02 ECC	172

Mini-cartridge (Irwin)	173
Using floppy disks	173
Formatting floppy disks	173
/etc/default/format file	174
Copying floppy disks	174
Using floppies for file storage	175
Making filesystems on floppy disks	176
Mounting a floppy filesystem	178

Chapter 8

Using MS-DOS and other DOS operating systems

179

OS/2 coexistence	180
Partitioning the hard disk using fdisk	180
Displaying current partition table	181
Switching operating systems	182
Installing a UNIX system partition on a DOS system	184
Using a UNIX system and DOS with two hard disks	185
Removing an operating system from the hard disk	185
DOS accessing utilities	186
File and directory arguments	186
User configurable default file	187
Mounting DOS filesystems on a UNIX system	187
Configuring support for mounted DOS filesystems	188
How DOS filesystems are accessed	188
Using the mount command	189
Mounting a floppy disk	189
Repairing and checking DOS filesystems	189
Who can access the mounted DOS filesystem	190
Appearance of DOS files	191
Restrictions	191
File names	191
Modification times	191
UNIX backup utilities	191

Index	997

Chapter 1

Introduction

The SCO® Open Desktop®/SCO Open Server™ operating system is SCO UNIX® System V/386. This guide is designed to cover tasks that the system administrator would need to perform to ensure the smooth and efficient running of the system. Your UNIX system requires careful control of its operation and a regular schedule of maintenance. This book explains how to run and maintain the operating system on your computer, ensuring maximum performance with a minimum number of problems.

NOTE The *Hardware Configuration Guide* explains how to connect and configure devices such as serial cards, hard disks, terminals, and tape drives. This guide is concerned with the use of such devices after they are installed. For example, the “Using floppy disks and tape drives” chapter of this guide discusses the use of magnetic storage media, and how to use a floppy or tape drive once it is installed.

The system administrator and administrative roles

Every UNIX system should have at least one person in charge of system maintenance and operation. In this guide, such a person is called a system administrator. It is the responsibility of system administrators to ensure the smooth operation of the system and to perform a wide variety of tasks that require special privileges.

You can choose to have a single system administrator or divide the tasks among several persons, each charged with a different area of operation. You can even assign roles that are strictly limited to one aspect of the system.

1. Introduction

Depending on the size of the system and the number of users on it, system administration can be anything from a once-a-day task to a full-time job. Even if the system is small, the system administrator should consistently perform each required maintenance task, because sloppy maintenance can adversely affect system performance.

The system administrator should keep a hard copy log of all system modifications and system events. Each event, message, backup, or modification should be logged with the date, time, and name of the person logging, and the circumstances surrounding the event. For example, if a new application is added to the system software, an entry should be placed in the log. This entry should include the time, date, and name of the person installing, and any notes about the software or installation that may be helpful. An accurate log helps in diagnosing system problems and charting the growth and use of a system.

All tasks in this guide are presented from a system administrator's point of view, but many can also be accomplished by ordinary users. Because some of the tasks dramatically change the system's operation, we recommend that, whenever possible, the system administrator perform these tasks. However, no matter who performs an operation, it should be entered in the system log. Following these rules can prevent unwanted or unnecessary changes to the system.

A system administrator has several tasks to perform, sometimes on a daily basis:

- Make certain the integrity of the system is not compromised through use of security mechanisms.
- Make certain that adequate backups (regular copies of files on the system) are made and stored for future use.
- Handle problems related to use of limited computer resources (disk space, number of processes, and so on).
- Alleviate system communication (network) stoppages due to failed connections.
- Apply operating system updates and maintenance fixes.
- Provide general support to users.

Making administration easier with the *sysadmsh*

The **sysadmsh**(ADM) command is a menu interface designed to simplify the task of system administration. The menus, submenus, and screens allow you to simply point and pick, or fill in blank fields. The **sysadmsh** allows less-experienced system administrators to use UNIX system commands that would otherwise require memorization and constant referring to manual pages. The **sysadmsh** includes context-sensitive help; simply press the (F1) key from any menu to display further explanations of the menu options.

If you are new to UNIX operating systems, we strongly recommend that you become familiar with the concepts and tasks covered in the material on the UNIX system in “Beneath the Desktop” in the *User's Guide*. This ensures that you can understand and perform the basic system administrative tasks described here.

To aid users of **sysadmsh**, the documentation of this guide is supplemented by **sysadmsh** references that appear below UNIX system command-line instructions.

For example, the following instructions refer to the **custom** utility, used to add more software to your system. Following the command is a sequence of **sysadmsh** menu selections.

Enter the following command:

custom

Δ **sysadmsh** users select: System ⇌ Software

This means that you can access the functions of the **custom** command by first double-clicking on the Administration icon in the Controls window, followed by selecting System at the main **sysadmsh** menu, followed by selecting Software at the next lower level. Selections can be made from the menu in any of the following ways:

- Move through the menu options using the (Space) key and press (Return) on the option you want.
- Move left and right through the options using the arrow keys and press (Return) on the required option.
- Press the first letter of the option required. This is the quickest way. Using the example above, you would simply enter **ss** (without the (Return) key) to reach the **custom** menu.
- Move through the menu options with a mouse, clicking the left button to select an option.

For more instructions on using the **sysadmsh**, refer to the “Using the system administration shell” chapter in this guide.

The super user account

The super user login (also known as *root*) is a special account for performing system maintenance tasks. It gives the system administrator unusual privileges that ordinary users do not have, such as accessing all files in the system, and executing privileged commands. Many of the tasks presented in this guide require that the system administrator be logged in as the super user. To do this, the system administrator must know the super user password created during the installation of your system. (See the *Installation and Upgrade Guide*.)

Log in as the super user only to perform system-maintenance tasks. Even if the system administrator is the only user of the system, that person should create a user account for day-to-day work, reserving the super user account for system-maintenance tasks only.

Few users should know the super user password. Misuse of the super user powers by naive users can result in a loss of data, programs, and even the operating system itself.

The keyboard

Many keys and key combinations perform special actions on UNIX systems. These actions have names that may not correspond to the keytop labels on your keyboard. Table 1-1 shows which keys on a typical terminal correspond to special actions on UNIX systems. A list for your particular login device is in `keyboard(HW)`. Many of these keys can be modified by the user; see `stty(C)`.

Table 1-1 Special keys

UNIX name	Action
<Return>	terminates a command line and initiates an action. This key is also called the <Enter> key; the keytop may indicate a down-left arrow.
<Esc>	exits the current mode; for example, exits insert mode when in the editor <code>vi</code> . This is also known as the ESCAPE key.
<Del>	stops the current program, returning to the shell prompt. This key is also known as the INTERRUPT key.
<Bksp>	deletes the character to the left of the cursor. The keytop may show a large left arrow, as opposed to the small "cursor left" arrow.

(Continued on next page)

Table 1-1 Special keys
(Continued)

UNIX name	Action
(Ctrl)d	signals the end of input from the keyboard; exits the current shell, or logs you out if the current shell is the login shell.
(Ctrl)h	deletes the first character to the left of the cursor. This is also called the ERASE key.
(Ctrl)q	restarts printing after it is stopped with (Ctrl)s.
(Ctrl)s	stops printing at the standard output device, such as a terminal. This keystroke does not stop the program.
(Ctrl)u	deletes all characters on the current line. This is also called the KILL key.
(Ctrl)\	quits current command, creates a <i>core</i> file. This is also called the QUIT key. (Use of this keystroke is recommended for debugging only; see <i>core(FP)</i> .)

Running programs simultaneously with MultiScreen

With MultiScreen™, you can run several programs on your console at the same time. You see the display for each program on a different “screen,” but you never have to leave your single console. Pressing a simple key combination switches you from one screen to another, and each screen acts independently from the others.

When you log in normally, you see the screen associated with the (Ctrl)(Alt)(F1) key combination. To open a second screen, press and hold the (Ctrl) and (Alt) keys, then press (F2) or another function key on your keyboard. Function keys are generally located across the top or down the far left side of your keyboard.

After you press the key combination, the screen clears and a login prompt appears. Log in again and you can begin work on the second “screen.” Press (Ctrl)(Alt)(F1) to switch instantly back to the first screen. Switching between screens in this way can speed up procedures that require working in two different programs. You can run both programs simultaneously, and work on either one at any time.

For example, you can start **sysadmsh** on your first screen, then press **<Ctrl><Alt><F2>** to create a second screen. Log in again and begin editing a log file with **vi**. Use **<Ctrl><Alt><F1>** and **<Ctrl><Alt><F2>** to switch back and forth between screens, taking actions in **sysadmsh** and recording them in the log.

You can open more than two multiscreens at once. However, make sure to log out of all extra screens before logging out of your first screen. Unattended screens where you remain logged in allow unauthorized access to the system.

If you have several screens open at once, you can rotate through them by pressing the Control and Print Screen key combination, **<Ctrl><PrtSc>**. This is helpful if you cannot remember which process is running on a particular screen.

For more information, refer to **multiscreen(M)** and **screen(HW)**.

System security

An important consideration is protecting the system and its data from unauthorized access. This system includes security mechanisms not found on other UNIX systems. These mechanisms are designed to meet the C2 class of “trust” as defined by the Trusted Computer System Evaluation Criteria (also known as the *Orange Book*). As the system administrator, you can configure the protection mechanisms to the requirements of your site. You can also set up the powerful auditing features to keep detailed records of logins and system usage. The “Administering user accounts” chapter explains how to add users to the system and how to configure the default security scheme. For additional information on using the security features, consult the chapters on system security and auditing in the *Open Server System Administrator’s Guide* in the Open Desktop Supplementary Documentation.

Site planning considerations

Before installing or expanding your system, review the following list of considerations with management and create a physical plan of the system that makes the most efficient use of available resources, and allows users the most direct and complete access to those resources.

If growth is expected, plan for it. Make sure that the resources and the communications lines that connect them are adequate for expected growth.

- Set up the computers in a place where they will not be bumped or moved at any time. If possible, they should be in a room by themselves, with little or no foot traffic. If workstations are used for data storage, at least place them on stable furniture and leave no cables exposed to traffic.

- Keep the computer room cool and give each machine excellent ventilation; keep all machines away from walls and, if possible, provide a separate air conditioner for the computer room, with more-than-adequate cooling capability.
- Install a Halon fire extinguishing system in the computer room rather than sprinklers.
- Store backup media in a separate room from the computers. This room should be fireproof, or should at least have a Halon fire extinguishing system rather than a sprinkler system.
- Ensure that there is adequate and uninterrupted power for the computers. This means more than enough current, and at least surge suppressors, if not a means of guaranteeing uninterrupted power supply. If power fluctuations and failures are common, provide a backup power supply. The computers should also be on an isolated, fully grounded (earthed) circuit.
- If you install a Local Area Network, plan the cabling and location of all machines and peripherals carefully. Seek the assistance of a networking expert to make these plans. Good planning and the use of adequate connecting media and compatible hardware are essential for long-term network performance.
- If you need modem lines for off-site connections, arrange for these with your local telephone company.
- If you plan to connect a printer to a parallel port, locate it close to the machine running it, but keep the machine out of the path of traffic to and from the printer.
- If you connect terminals, printers, or other peripherals to serial ports, consider using phone-line cabling and switching hardware, especially if your system is expected to grow in size and complexity. You can readily adapt phone lines for serial hardware, and telephone connecting and switching technology is mature and flexible.

Chapter 2

Using the system administration shell

The **sysadmsh** (system administration shell) is a menu interface designed to simplify the task of system administration.

You will find it easier to learn the material in this chapter if you start the **sysadmsh** and actually run the examples as you get to them.

This chapter assumes that you have some knowledge of the UNIX operating system. You should become familiar with the concepts covered in the material on the UNIX system in “Beneath the Desktop” in the *User’s Guide*, before using the **sysadmsh** options.

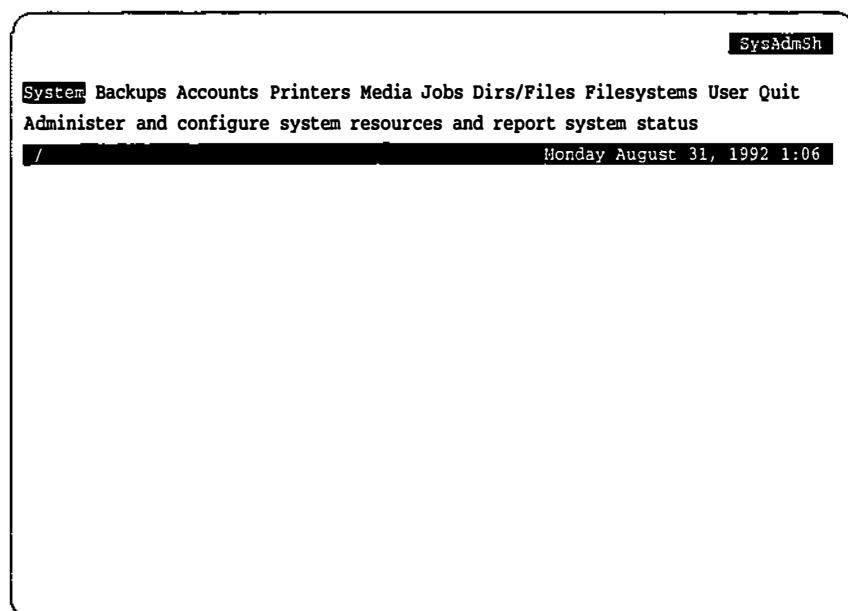
Starting sysadmsh

To gain access to all the functionality of **sysadmsh**, double-click on the Administration icon in the Controls window.

NOTE Only system administrators can make most selections from **sysadmsh**, and only the *root* account can make all selections. If you have insufficient authorization to perform an action (such as backups) you see an error message stating that access is denied because an authorization is required. Refer to “Subsystem authorizations” (page 91) for more information.

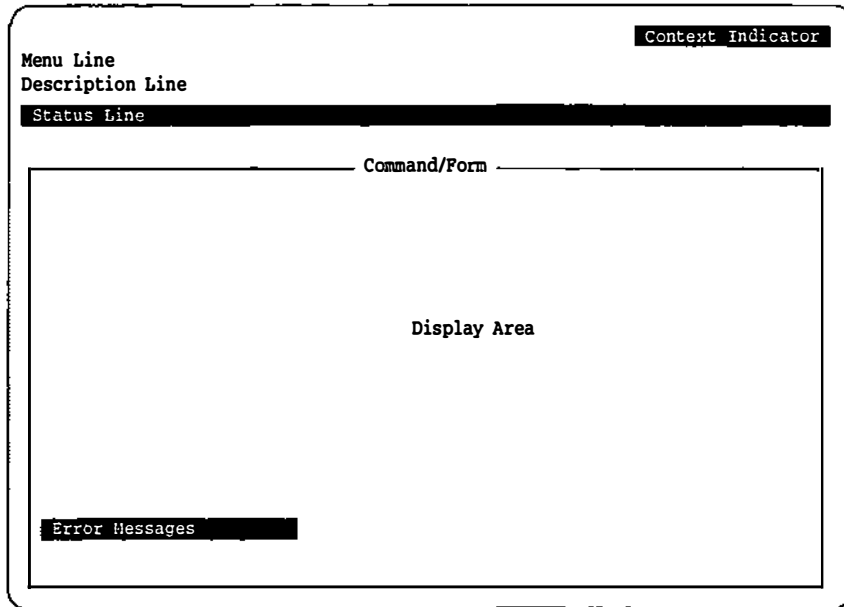
2. Using the system administration shell

The main `sysadmsh` menu is displayed:



How the screen is organized

This is a schematic of the **sysadmsh** screen. Areas shown in black appear on the screen as highlighted areas or bars of text. Each area displays specific types of information:



- The Context Indicator is the highlighted bar of text in the upper-right corner of your screen. It displays the name of the current menu. The Context Indicator for the **sysadmsh** opening screen shows *SysAdmSh*.
- The Menu Line displays the menu options that are currently available. The main **sysadmsh** menu consists of ten options: System, Backups, Accounts, Printers, Media, Jobs, Dirs/Files, Filesystems, User, and Quit.
- The Description Line gives you a brief description of the currently highlighted menu option.
- The Status Line is the highlighted bar of text that separates the Menu and Description Lines from the Display Window. The Status Line in the **sysadmsh** opening screen contains the date, time, and current working directory. When a UNIX system command is executed, the name of the command and the options used are displayed briefly at the far left of the Status Line.

- The Command/Form Line displays a title for the contents of the Display Area. The title can be either a UNIX system command name or the name of a **sysadmsh** form. When a command name is displayed, the location of the manual page associated with the command is appended in parentheses. For example, when System ⇌ Report ⇌ Users is selected, the Command Line displays "who C". This means that the command can be found on the **who(C)** manual page.
- The Display Area displays **sysadmsh** forms and scan windows. Forms and scan windows are explained in detail later in this chapter.
- Error Messages and recovery instructions appear on the last line of the screen in highlighted text.

Selecting menu items

The keyboard and mouse operations listed in Table 2-1 are used to move through the menus. Note that there are several ways to select options; if you have used menu-based programs before, use the method you are most familiar with.

Table 2-1 Basic menu operations

Keyboard	Mouse	Action
Arrow keys or <Space> (same as right arrow)	Move cursor to option	moves to menu option.
First letter of option, or move highlight to option and press <Return>	Left or middle button	selects menu option.
<Esc>	Right button	retreats to previous menu.
<F1>	—	gets help.

You can familiarize yourself with the menu options by using the Arrow keys, <Space>, or mouse to move the highlight from option to option. Each time you move the highlight to a new option, a description of that option appears on the description line.

sysadmsh has a hierarchical menu structure. Many of the menu options move you down to another menu. For example, when you select the Jobs option from the main menu, a submenu containing more options is displayed which lets you check on and manipulate your machine's processes. The menu hierarchy makes it easy to find the command you need by moving down from one menu to the next. Eventually you get to a menu option that either executes a UNIX system command or displays a form that you must fill in with the details that the command needs. Note that typing the first letter of the

option name is the quickest way to move through menu levels; in time you will be able to reach the function you need instantly by pressing three- and four-letter codes you have memorized. (Table 2-9 at the end of this chapter lists all available `sysadmsh` selections and their shorthand forms.)

The best way to learn how to use menus is to practice making menu selections with the keyboard or mouse. If you select an option by mistake, you can always retreat to the previous menu by pressing the (Esc) key or the right mouse button. If you are several levels deep, you can return to the main menu by pressing the (F2) key and then typing `n`. (F2) takes you to the Quit option, and `n` returns you to the main menu. (The (F2) key does not work if you have been dropped into a UNIX system command, such as `vi(C)`.) To help you find your way through the `sysadmsh` menus, Table 2-2 contains a map of the second-level menus.

Table 2-2 Map of second-level menus

System ↓	Backups ↓	Accounts ↓	Printers ↓	Media ↓
Report	Create	User	Configure	List
Configure	Restore	Defaults	Schedule	Extract
Hardware	Schedule	Terminal	Request	Archive
Software	View	Report	Auxiliary	Format
Audit	Integrity	Check	Priorities	Duplicate
Execute				Tapedump
Terminate				
Jobs ↓	Dirs/Files ↓	Filesystems ↓	User*	Quit ↓
Report	List	Check		Yes
Terminate	View	Mount		No
Authorize	Copy	Unmount		
	Edit	Add		
	Modify	Floppy		
	Print	DOS		
	Archive			
	Differences			
	Remove			
	UseDOS			

* This selection can be used for user-specific applications defined by the `SA_USERAPPS` environment variable or read from `$HOME/.sysadmmenu` (see `sysadmsh(ADM)` and `sysadmmenu(F)` for more information).

When you select a menu option, one of three things happens:

- A lower-level menu is displayed.
- You are dropped into a form.
- A UNIX system command is executed and the result displayed in a scan window.

The next two sections explain forms and scan windows.

Using forms

Some menu options require additional information to perform the correct task. For example, the Print option cannot do anything until you tell it what you want to print and which printer to use. When you select this type of option, a form appears on the screen. By filling in the form, you give the command the information it needs.

The following example demonstrates how forms work, by showing you how to print a file in your current directory. After the example, Tables 2-3, 2-4, and 2-5 list the keystrokes that allow you to move around the form, edit it, and make “point-and-pick” selections.

To print a file, first select Dirs/Files ⇌ Print . The Print form is displayed:

The screenshot shows a terminal window with a form titled "Print Files". At the top, it says "Enter file or directory name or press (F3) for a file list" with a "Print" button to the right. Below this is a status bar showing a slash "/" and the date/time "Monday August 31, 1992 1:06". The main area of the form contains two input fields: "Enter file(s) to print:" and "Enter destination printer:", each followed by a pair of square brackets for text entry.

```
Enter file or directory name or press (F3) for a file list Print  
  
/ Monday August 31, 1992 1:06  
  
----- Print Files -----  
  
Enter file(s) to print: [ ]  
Enter destination printer: [ ]
```

Notice that the highlight is on the first item in the form. You can fill in the field or obtain a list of choices by pressing (F3). You can enter the filename if you know it but, for the sake of this exercise, assume that you need to find the filename and press (F3) now. A window opens up overlapping part of the Print form:

```

Enter file or directory name or press (F3) for a file list
/ Monday August 31, 1992 1:06

Print Files

Enter file(s) to print:  [
Enter destination printer: [

file1  file2  file3  file4
file5  file6


```

The window contains a list of the files that you can select. To select a file, “point” to it by highlighting it, and “pick” it by pressing return. This is known as point-and-pick, and it is used whenever a range of choices is displayed. After making your selection, the window closes and you return to the Print form.

Note that the name of the file you selected is now displayed in the form. You can now change the name using the edit keys (listed in Table 2-4 later in this section), or press (Return) to move to the next field.

2. Using the system administration shell

Now enter the name of the printer to be used. If you do not know the printer name, press (F3). Another, smaller window opens that contains a list of installed printers:

The screenshot shows a terminal window titled "Enter printer name or press (F3) for a list of printers". The window has a status bar at the top right showing "Monday August 31, 1992 1:06". Inside the window, there is a "Print Files" dialog box. The dialog box has two input fields: "Enter file(s) to print: [file1]" and "Enter destination printer: [". A small window is open over the printer selection field, displaying a list of printers: "printer1", "printer2", and "printer3". The "printer3" option is highlighted with a black background.

You can select the printer just as you did the name of the file. After selecting a printer, you return to the Print menu.

The keystrokes listed in the following tables allow you to use forms easily.

Table 2-3 Form operations

Keyboard	Mouse	Action
<Esc>	Right button	tells the program that you changed your mind and do not want to finish filling in this form. The form is removed, and no action is performed. You are returned to the previous menu. In addition, <Esc> followed by <Return> acknowledges that an error message was read and that you are ready to continue.
Up, Down Arrow	Left button	moves to other fields in a form. Some fields are restricted and no input is allowed. The Arrow keys skip over these. Other fields must be filled in. With the mouse, move the cursor to the field and click the Left button. Pressing the Down Arrow key on the last item in a form brings you back to the first item.
Left, Right Arrow	—	moves left and right in the current field. This allows you to change text without retyping the entire line.
<Return>	Middle button*	completes the data entry to a field and moves the cursor to the next field. In the last field, pressing <Return> or the middle mouse button completes the entire form and tells the shell that the data is ready to use.
<Ctrl>x	—	exits and executes the form from wherever you are. Think "x" for "execute". <F10> does the same.
<F4>	—	calls the spelling checker when you are in a form. If you think a word might be misspelled, press <F4> while the cursor is on the word and a list of possible correct spellings appears in a point-and-pick list. The word you select replaces the misspelled word.
<F10>	—	exits and executes the form from wherever you are. <Ctrl>x does the same.

* On a two-button mouse, pressing both buttons simultaneously emulates the middle mouse button.

2. Using the system administration shell

Table 2-4 Edit keystrokes*

Keystroke	Action
<Ctrl>y	deletes the current line, and begins the line again.
<Ctrl>w	deletes the current word.
<Ctrl>g-<Ctrl>h	moves the cursor to the beginning of the line.
<Ctrl>g-<Ctrl>l	moves the cursor to the end of the line.
<Ctrl>v	toggles into or out of overstrike mode.
<Del>	deletes the character over the cursor.
<Bksp>	backs up and deletes one character (left of cursor).
<Ctrl>u	pages up — moves up one page.
<Ctrl>d	pages down — moves down one page.
<Ctrl>n	goes to the next word.
<Ctrl>p	goes to the previous word.
Left, Right Arrow	moves left and right within the edit line.

* You cannot use the mouse for edit operations; if a form opens a window for input, you must use the keyboard. Even the Rightbutton (<Esc>) will not affect the window.

NOTE The edit keystrokes can also be used to edit the name of the current directory (accessed using <F6>).

Table 2-5 Point-and-Pick operations

Keyboard	Mouse	Action
<Return>	Middle button*	selects the item.
<Esc>	Right button	ends the selection process. The list is removed and no action is performed.
<Ctrl>v	—	toggles between selecting all or none of the items appearing in a list.
Up, Down Arrow	Move mouse up or down	moves to other items in a list.
Left, Right Arrow	Move mouse right or left	moves across a multicolumn display.
<Space>	Left button	marks items when the application accepts more than one. A marked item is indicated by an asterisk (*) in the left column. It may be unmarked by pressing <Space> a second time while the item is selected. The entire collection of marked items is selected by pressing <Return>.
<F5>	—	finds items in long listing. It is called the "Search" key. A prompt appears and you enter the string to search for, then press <Return>. If the item is found, the highlight moves to that item, and another <Return> selects the item. If no match is found, the highlight does not move. The ";" and ":" keys repeat the previous search, forward and backward respectively.
First letter	—	selects an item by its first letter. It is the fastest method of selection. Pressing <Return> selects the highlighted item. (If there is only one item beginning with that letter, it is marked by typing its first letter. There is no need to press <Return> again.) If several items begin with the same letter, the cursor moves to the first occurrence in the list.

* On a two-button mouse, pressing both buttons simultaneously emulates the middle mouse button.

Using radio buttons

Radio buttons are rows of selection boxes in a form. They are selected in a way similar to items in a menu bar. The purpose of radio buttons is to provide a secondary level of selections from within a form, as in this example, taken from the Accounts menu of the **sysadmsh**:

The screenshot shows a terminal window with the following content:

```
Name of an existing user (<F3> for list) [redacted]
```

```
/tmp Monday August 31, 1992 1:06
```

```
View/Modify an existing user's account
```

```
Username : [      ]
```

```
[ Audit ] Expiration Identity Logins Password Privileges
```

In this example, the “Username” field must be filled in before the radio button selection can be made. The reference syntax for **sysadmsh** includes a notation for radio buttons, which is a colon (:). Look at the following reference:

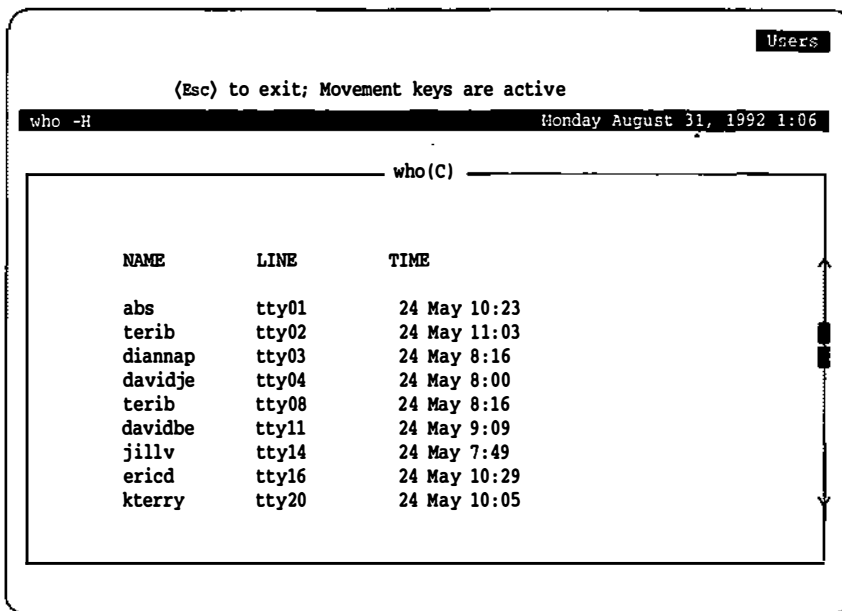
Accounts ⇌ User ⇌ Examine:Audit

When you see a **sysadmsh** reference like this, it means that when you enter the Examine menu you must provide information (in this case, the user name) before making the next selection: Audit.

Using scan windows

When you execute a UNIX system command by selecting a **sysadmsh** menu option, the result of the command is typically displayed in a scan window. Scan windows also display the contents of files and directory listings. To demonstrate the use of scan windows, let's say you want to know who is currently logged on to the system. To do this, make the selection: System ⇨ Report ⇨ Users. (This runs the UNIX system **who(C)** command.)

When you select the Current option, a scan window displaying the output of the **who(C)** command appears in the display area:



Note that the name of the command **who** and the reference section in which its description can be found (C) are displayed at the top of the window. Also note that the option given to the command (-H) is displayed in the left hand side of the Status Line. If you do not understand the information displayed, look up the proper manual page for more information.

The vertical scroll bar (at the extreme right edge of the window) shows the position of the current screen relative to the whole document, and moves the screen up or down by an amount that depends on how far you move the scroll bar. When you are viewing the first screen in a document, the top arrow is covered by the highlighted block; when you are viewing the last screen in a document, the bottom arrow is covered by the highlighted block. If the document is less than one page long, the scroll bar does not appear.

To operate the scroll bar, press and hold the <Down Arrow> or <Up Arrow> keys, depending on which direction you want the screen to move over the page. When the scan window shows the area of the document that you want to view, release the key. The <PgUp> and <PgDn> keys can be used to move over the document a page at a time.

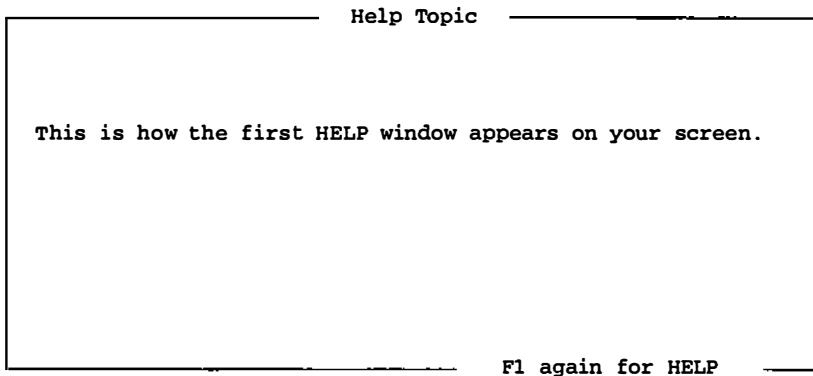
Use the keys listed in Table 2-6 when you are in a scan window.

Table 2-6 Scan operations

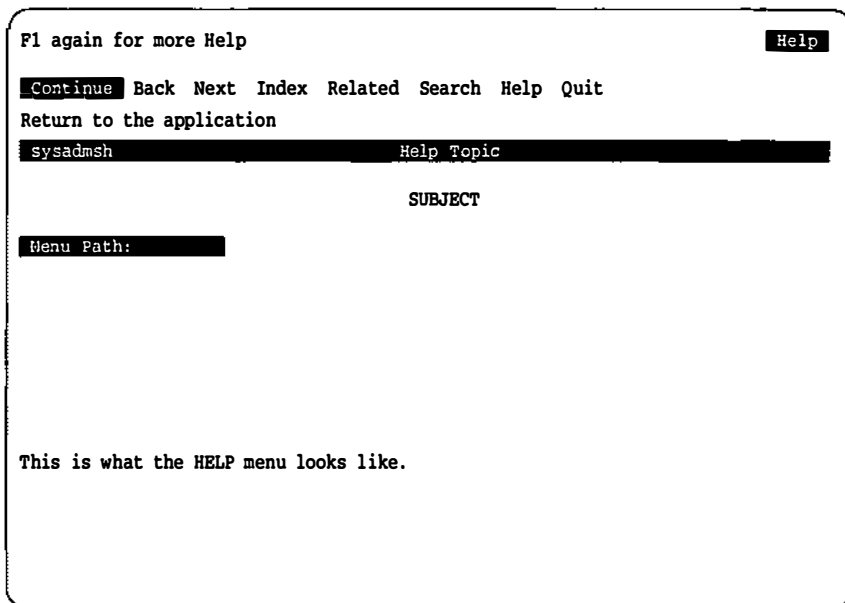
Keyboard	Mouse	Action
<Esc>	Right button	exits the file.
<Up Arrow>	—	moves up one line.
<Down Arrow> or <Return>	—	moves down one line.
<PgDn> or <Space>	Middle button	moves down a page.
<PgUp>	Left button	moves up a page.
<Home>	—	moves to the top of the display.
<End>	—	moves to the bottom of the display.
<F5>	—	searches for a pattern in the display. (The “;” and “:” keys repeat the search forward and backward, respec- tively.)
<F7>	—	prints the output of the command or file currently in the scan window.

Getting help

You can press the (F1) key to display more information to help you with your selection. When you press the (F1) key, a Help window opens within your current screen. It looks like this:



The window contains some basic information. If you need more help, you can press (F1) again and the complete Help menu is displayed:



When you are finished, select Quit from the Help menu and you return to your place in the `sysadmsh` menu.

The menu options for Help are listed in Table 2-7.

Table 2-7 Help options

Option	Action
Continue	continues on to the next page of help text. All the vertical movement keys are active: Up and Down Arrows, (PgUp) and (PgDn), (Home) and (End). If there is no further information, the highlight moves to the Quit option on the Help menu and the description line reads "Return to the application".
Back	moves back to topics that were seen previously. There is no corresponding "Forward". This also backs up to more general topics. You can go back until the top-level introductory topic is reached.
Index	chooses a new topic from a list of indexed topics.
Related	chooses a new topic related to the current one.
Search	searches for a new topic by matching a pattern. First, you specify where to look (the titles, the text lines, or both), and then give the pattern. The pattern can be a simple keyword (like "create" or "date") or a more complex <i>regular expression</i> . A list of topics containing the pattern is presented.
Help	explains how the help facility itself is used. A table similar to this one is displayed on the screen. If you need further information, look for your topic in Index, Related, or Search.
Quit	exits Help and returns to <code>sysadmsh</code> . (F2) or (Esc) are other ways to exit quickly.

Each Help screen has general information available, as well as specific information about each option listed on the menu from which Help was selected. Each descriptive passage is preceded by the associated menu line and followed by a reference to the operating system documentation.

NOTE When you are within a particular UNIX system command, you do not have access to the Help facility. For example, when you select Dirs/Files ⇨ Edit, you are within the UNIX system vi command, and the **sysadmsh** keys no longer function. When you exit the command and return to the **sysadmsh**, the keys function as expected. If no element of the **sysadmsh** is visible on the screen (menu line, boxes, context indicator, and so on) then Help is probably unavailable. If you need help, exit from the current process and press the (F1) key to view Help. In general, it is best to use Help prior to executing a menu selection.

Changing the current directory within sysadmsh

There are many occasions when it is necessary to change your current directory to use certain files and commands. You can move to another directory by pressing the (F6) key. The current directory is displayed at the top of the screen. You can use the (Bksp) key to erase the name of the current directory (to begin again), or you can add to or alter part of the current name using the edit keystrokes (Table 2-2). When you press (Return), your directory change is executed and reflected on the status line.

The function keys

The function keys give you access to several time-saving features.

Table 2-8 Function keys

Key	Action
(F1) Help key	displays help for the current context within the application. Further information is available by pressing (F1) again.
(F2) Exit key	activates the Quit option on the top menu level. Press n to return to sysadmsh or y to exit.
(F3) Pop-up key	(used within a form) displays a list of items that are acceptable for the current field.
(F4) Spell key	(used within a form) displays a list of words that are possible correct spellings of the word in the current field. Select a word from the list by highlighting it, then pressing (Return). The word is then placed in the field. This function is only available if the Speller is installed.

(Continued on next page)

Table 2-8 Function keys
(Continued)

<F5> Search key	(used within a window) prompts for a string to search for. When you enter a string and press <Return>, the highlight moves to the item in the list that matches the pattern. If no match is found, the search fails and the highlight does not move. In addition, the semicolon (;) repeats a search forward and the colon (:) searches backward.
<F6> New directory key	offers the opportunity to change your current working directory. Note that this does not change the directory you will return to upon leaving sysadmsh .
<F7> Print key	prints the output of any command that is displayed in a scan window.

Using shell escapes to access the UNIX system command line

You can execute a UNIX system command from within a **sysadmsh** menu by typing the shell-escape character, an exclamation point or mark (!). The menus are replaced by a subshell that displays a text-entry line and a prompt asking for a command. When you enter the command and press <Return>, the command is executed by the shell. After the command is completed, the output is displayed on the screen, and you are prompted to press any key to return to the shell.

NOTE The UNIX system command line can only be accessed from the shell menus. It cannot, for example, be accessed from a form or a point-and-pick list.

Customizing sysadmsh menus

Third party developers can customize **sysadmsh** menus as desired, including changing colors. See **menumerge(ADM)** and **sysadmcolor(F)** for complete instructions.

sysadmsh menu options

Table 2-9 lists the options available in the **sysadmsh** menus. The main or top-level menu options appear in boldface type at the left margin of the table. The options listed below each top-level menu option are in lower levels of the menu hierarchy. Those with one arrow are in the first-level menus, those with two arrows are in the second-level menus, and so on. For instance, at the beginning of the table, Report is a first-level menu option in the System menu, and Activity is an option in the second-level Report menu.

Once you are familiar with the menu options, you can enter shorthand menu paths to reach lower level options without having to see all of the upper level menus. In Table 2-9, the "Path" column lists the shorthand menu path for each menu option. Enter the first letter of each menu option in the path down to the option you want. For example, to choose the Activity option, enter the first letters of system, report, and activity, *sra*, without pressing the (Return) key.

If there are no menu options listed below the one you choose, either a form or a display appears. For example, when you choose the Activity option, a scrollable display or scan window appears, showing system processes.

Table 2-9 sysadmsh menu map

sysadmsh option	Path	Function
System	s	system-wide reports, configurations
⇒ Report	sr	reports on current state of system
⇒⇒ Activity	sra	reports on current system activity
⇒⇒ Users	sru	reports users currently logged in
⇒⇒ Printers	srp	reports on current status of printers
⇒⇒ Disk	srd	reports on current disk usage
⇒⇒ Network	srn	reports on current network files
⇒⇒⇒ Xnet	srnx	reports on XENIX-NET status
⇒⇒⇒ UUCP	srnu	reports on UUCP status
⇒⇒ Messages	srn	reads system messages
⇒⇒ Software	srs	checks installed software/package status
⇒ Configure	sc	configures system files
⇒⇒ Security	scs	changes level of system security
⇒⇒⇒ Relax	scsr	resets security to default UNIX system levels
⇒⇒ Kernel	sck	changes kernel parameters or capabilities
⇒⇒⇒ ParaMeters	sckp	configures tunable kernel parameters
⇒⇒⇒ Rebuild	sckr	relinks kernel according to current settings

(Continued on next page)

Table 2-9 sysadmsh menu map
(Continued)

sysadmsh option	Path	Function
⇔⇔⇔DOS	sckd	adds DOS filesystem support to the kernel
⇔⇔⇔Streams	scks	adds streams support to the kernel
⇔⇔⇔Layers	sckl	adds shell layers support to the kernel
⇔⇔Logout	scl	sets idle time before user is logged out
⇔⇔Defaults	scd	modifies system default parameters
⇔⇔⇔Home	scdh	modifies default home directory for users
⇔⇔⇔Message	scdm	modifies message of the day file
⇔⇔⇔Checklist	scdc	lists which filesystems checked at startup
⇔⇔⇔Other	scdo	modifies other default files
⇔⇔International	sci	configures system for international use
⇔⇔⇔System	scis	sets system locale variables
⇔⇔⇔Individual	scii	changes a user's default locale
⇔⇔⇔Display	scid	changes mapping of terminal character set
⇔⇔⇔Keyboard	scik	changes mapping of console keyboard
⇔⇔Network	scn	configures networking files
⇔⇔⇔UUCP	scnu	configures UUCP files, enable/disable tty
⇔⇔Time	sct	sets the system time
⇔⇔Menus	scm	customizes sysadmsh menus
⇔⇔Other	sco	executes third-party sysadmsh extensions
⇔Hardware	sh	adds or removes hardware from the system
⇔⇔HardDisk	shh	adds a hard disk to the system
⇔⇔Tape	sht	adds or removes a tape drive from system
⇔⇔Printer	shp	adds or removes a printer from system
⇔⇔Card_Serial	shs	adds a serial card to system
⇔⇔Mouse	shm	adds mouse to system
⇔⇔Video	shv	configures video card graphics parameters
⇔Software	ss	adds or removes software from system
⇔Audit	sa	administers/examines system auditing data
⇔⇔Enable	sae	enables audit using existing parameters file
⇔⇔Disable	sad	disables audit (stops collection of process data)
⇔⇔Collection	sac	displays or modify audit collection rules
⇔⇔⇔Directories	sacd	displays or modify audit directory list
⇔⇔⇔⇔List	sacdl	displays audit directory list
⇔⇔⇔⇔Create	sacdc	creates new audit directory
⇔⇔⇔⇔Delete	sacdd	deletes existing audit directory
⇔⇔⇔⇔Add	sacda	adds entry to audit directory list

(Continued on next page)

Table 2-9 sysadmsh menu map
(Continued)

sysadmsh option	Path	Function
⇨⇨⇨⇨Remove	sacdr	removes entry from audit directory list
⇨⇨⇨Events	sace	displays/modifies list of audited events
⇨⇨⇨View	sacev	displays system audit collection mask
⇨⇨⇨Modify	sacem	modifies system audit collection mask
⇨⇨⇨IDs	saci	displays/modifies list of users/groups audited
⇨⇨⇨View	saciv	displays list of users and groups audited
⇨⇨⇨Modify	sacim	modifies list of users and groups audited
⇨⇨⇨Parameters	sacp	displays or modifies audit parameters
⇨⇨⇨View	sacpv	displays audit parameters
⇨⇨⇨Modify	sacpm	modifies audit parameters
⇨⇨⇨Reset	sacr	changes collection rules to default values
⇨⇨⇨Statistics	sacs	displays statistics of current audit session
⇨⇨⇨Report	sar	reports on stored audit session data
⇨⇨⇨List	sarl	lists report templates available
⇨⇨⇨View	sarv	views parameters in a report template
⇨⇨⇨Create	sarc	creates a new report template
⇨⇨⇨Modify	sarm	modifies an existing report template
⇨⇨⇨Delete	sard	deletes an existing report template
⇨⇨⇨Generate	sarg	generates report of an audit session
⇨⇨⇨Files	saf	manipulates audit session files
⇨⇨⇨List	safl	lists audit session files on system
⇨⇨⇨Backup	saftb	backs up audit session file to removable media
⇨⇨⇨Delete	saftd	removes audit session file
⇨⇨⇨Restore	saftb	restores audit file from removable media
⇨Execute	se	executes programs that are system specific
⇨Terminate	st	shuts down system to remove power or reboot
Backups	b	performs backups of files
⇨Create	bc	creates backups
⇨⇨Scheduled	bcs	performs scheduled filesystem backups
⇨⇨Unscheduled	bcu	performs unscheduled filesystem backup
⇨Restore	br	restores filesystems and files
⇨⇨Partial	brp	restores specific directories and files
⇨⇨Full	brf	restores entire filesystem

(Continued on next page)

Table 2-9 sysadmsh menu map
(Continued)

sysadmsh option	Path	Function
⇨Schedule	bs	modifies scheduled backup frequency
⇨View	bv	views contents of a backup
⇨Integrity	bi	checks the integrity of a backup
Accounts	a	controls functions of user accounts
⇨User	au	alters/creates/retires user accounts
⇨⇨Examine	aue	sets password/ID/authorization/audit parameters
⇨⇨Create	auc	makes a new user account
⇨⇨Retire	aur	closes an existing user account
⇨Defaults	ad	sets system-wide (default) parameters
⇨⇨Authorization	ada	views/modifies default kernel/subsystem privileges
⇨⇨Password	adp	views/modifies default password life/choice
⇨⇨Logins	adl	views/modifies default login attempt controls
⇨Terminal	at	manages terminal database entries
⇨⇨Examine	ate	views/modifies existing terminal entry
⇨⇨Create	atc	makes a new terminal entry
⇨⇨Delete	atd	deletes an existing terminal entry
⇨⇨Lock	atl	locks a specific terminal
⇨⇨Unlock	atu	clears all locks on a specific terminal
⇨⇨Assign	ata	manages device name equivalences database
⇨⇨⇨Examine	atae	views/modifies existing device entry
⇨⇨⇨Create	atac	makes a new device entry
⇨⇨⇨Delete	atad	deletes an existing device entry
⇨Report	ar	reports on passwords/terminals/login activity
⇨⇨Password	arp	reports on accounts by password status
⇨⇨⇨Impending	arpi	reports on accounts: passwords near expiring
⇨⇨⇨Expired	arpe	reports on accounts with expired passwords
⇨⇨⇨Dead	arpd	reports on accounts with dead passwords
⇨⇨⇨User	arpu	reports on single user's password
⇨⇨⇨Group	arpg	reports on single group's passwords
⇨⇨⇨Full	arpf	lists all entries in password database
⇨⇨Terminal	art	reports on access status by terminals
⇨⇨Login	arl	reports login activity by user/group/terminal

(Continued on next page)

Table 2-9 sysadmsh menu map
(Continued)

sysadmsh option	Path	Function
⇔⇔⇔User	arlu	reports logins of one, a range of, or all users
⇔⇔⇔Group	arlg	reports logins of a group or range of groups
⇔⇔⇔Terminal	arlt	reports logins of one/range/all terminals
⇔Check	ac	checks contents of tcb files for errors
⇔⇔Databases	acd	checks consistency of subsystem databases
⇔⇔Password	acp	checks <i>/etc/passwd</i> and <i>/etc/group</i>
Printers	p	administers print system
⇔Configure	pc	configures printers on print service
⇔⇔Add	pca	adds printer to system
⇔⇔Modify	pcm	modifies printer configuration
⇔⇔Remove	pcr	removes printer destination from print service
⇔⇔Default	pcd	changes system default destination printer
⇔⇔Parameters	pcp	modifies printer controls and parameters
⇔⇔Errors	pce	sets error warning notification/recovery modes
⇔⇔Content	pcc	specifies type of content printable on printer
⇔⇔Users	pcu	specifies who can use printer
⇔Schedule	ps	starts/stops print service, handle requests
⇔⇔Begin	psb	starts print service
⇔⇔Stop	pss	shuts down print service
⇔⇔Accept	psa	allows requests for destination
⇔⇔Reject	psr	rejects requests for destination
⇔⇔Enable	pse	enables printers
⇔⇔Disable	psd	disables printers
⇔Request	pr	moves or cancels requests on print service
⇔⇔Move	prm	moves requests between destinations
⇔⇔Cancel	prc	cancels requests made to print service
⇔Auxiliary	pa	handles print-wheels/filters/preprinted forms
⇔⇔Alert	paa	sets/lists an alert for print-wheel
⇔⇔Filter	paf	administers filters used with print service
⇔⇔⇔Change/Add	pafo	adds or changes filter used with print service
⇔⇔⇔Remove	pafr	removes filter from print service
⇔⇔⇔List	pafl	lists a description of filter
⇔⇔⇔Original	pafo	restores original filter description

(Continued on next page)

Table 2-9 sysadmsh menu map
(Continued)

sysadmsh option	Path	Function
⇔⇔PPforms	pap	administers pre-printed forms of print service
⇔⇔⇔Configure	papc	modifies printer settings for pre-printed forms
⇔⇔⇔Modify	papm	adds or changes pre-printed forms
⇔⇔⇔Remove	papr	removes pre-printed form from print service
⇔⇔⇔List	papl	lists attributes of existing form
⇔⇔⇔Users	papu	allows or denies user access to a form
⇔⇔⇔Alerts	papa	modifies alert method for mounted form
⇔⇔⇔⇔Specify	papas	specifies alerting method
⇔⇔⇔⇔List	papal	lists current alert
⇔⇔⇔⇔Terminate	papat	terminates existing active alert
⇔⇔⇔⇔Remove	papar	removes alert definition
⇔Priorities	pp	sets printing queue priorities
⇔⇔Default	ppd	sets system-wide priority default
⇔⇔Highest	pph	sets default highest priority level for users
⇔⇔Remove	ppr	removes users from specified priority level
⇔⇔List	ppl	lists default priority level and limits
Media	m	reads, copies, compares, formats floppies/tapes
⇔List	ml	lists the contents of a floppy or tape
⇔Extract	me	extracts the contents of a floppy or tape
⇔Archive	ma	stores files/directories/filesystems on media
⇔Format	mf	formats either a UNIX or DOS floppy
⇔Duplicate	md	makes a copy of floppy or tape
⇔Tapedump	mt	displays physical contents of tape
Jobs	j	views/controls processes
⇔Report	jr	reports on current processes (snapshot)
⇔Terminate	jt	terminates currently running process (kill)
⇔Authorize	ja	authorizes users to run jobs
⇔⇔Scheduled	jas	authorizes creating regularly scheduled jobs
⇔⇔⇔Default	jasd	sets default authorization for scheduled jobs
⇔⇔⇔User	jasu	allows/prohibits user-scheduled jobs
⇔⇔⇔View	jasv	checks who can create scheduled jobs
⇔⇔Delayed	jad	authorizes users to create delayed jobs
⇔⇔⇔Default	jadd	sets the default authorization for delayed

(Continued on next page)

Table 2-9 sysadmsh menu map
(Continued)

sysadmsh option	Path	Function
⇔⇔⇔User	jadu	jobs allows/prohibits user from creating delayed jobs
⇔⇔⇔View	jadv	checks who is authorized to create delayed jobs
⇔⇔Environment	jae	modifies delayed job environment
⇔⇔⇔At	jaea	modifies the environment for at-controlled jobs
⇔⇔⇔Batch	jaeb	modifies the environment for batch(C) jobs
Dirs/Files	d	interacts with files and directories
⇔List	dl	lists files in current directory
⇔View	dv	views contents of file
⇔Copy	dc	copies directory or file
⇔Edit	de	edits one or more files
⇔Modify	dm	changes file parameters
⇔⇔Permissions	dmp	changes file permissions
⇔⇔Ownership	dmo	changes file ownership
⇔⇔Group	dmg	changes file group ownership
⇔⇔Name	dmn	renames or moves files
⇔⇔Size	dms	compacts files
⇔⇔Format	dmf	changes file formats
⇔Print	dp	prints files
⇔Archive	da	stores files
⇔Differences	dd	compares two text files or directories
⇔Remove	dr	removes specified files or directories
⇔UseDOS	du	uses DOS utilities to manipulate DOS files
⇔⇔List	dul	lists DOS files in current directory
⇔⇔Remove	dur	removes a DOS file or directory
⇔⇔MakeDir	dum	creates a DOS directory
⇔⇔Copy	duc	copies files between DOS and UNIX systems
⇔⇔View	duv	displays DOS files
⇔⇔Format	duf	formats DOS media
Filesystems	f	checks/acts on filesystems
⇔Check	fc	checks and repairs filesystem (fsck)
⇔Mount	fm	mounts a filesystem

(Continued on next page)

Table 2-9 sysadmsh menu map
(Continued)

sysadmsh option	Path	Function
⇨Unmount	fu	unmounts a mounted filesystem
⇨Add	fa	adds appropriate info for a new filesystem
⇨Floppy	ff	creates a filesystem on a floppy
⇨DOS	fd	adds support for DOS filesystem
User	u	user-specific applications read from <i>\$HOME/sysadmmenu</i>
Quit	q	quits system administration shell
⇨Yes	qy	leaves the sysadm shell
⇨No	qn	cancels Quit command

Chapter 3

Starting and stopping the system

This chapter explains how to do the following:

- Start and stop your system.
- Log in as the super user (*root*).
- Change the system boot procedure.
- Use the device and system configuration information displayed at boot time.

Starting the system

Starting a UNIX system requires more than just turning on the power. You must also perform a series of steps to initialize the system for operation. To start the system, you must do the following:

- Load the operating system.
- Check the filesystems (if the system was improperly stopped).
- Choose the mode of system operation.

The following sections describe each of these procedures.

Loading the operating system

The first step in starting the system is to load the operating system from the computer's hard disk.

1. Turn on power to the computer and hard disk. The computer loads the UNIX system bootstrap program and displays this message:

```
SCO System V/386
Boot
:
```

2. Press the <Return> key. The bootstrap program loads the operating system.

When the system is loaded, it displays information about itself and verifies that the root filesystem (that is, all files and directories) is in order and not corrupted. If a filesystem is uncorrupted and in good order, it is called "clean". If the root filesystem is clean, you can choose the mode of operation. If not, the system requires you to clean the filesystem before choosing.

Cleaning filesystems

You must clean the root filesystem if the following message is displayed:

```
fsstat: root filesystem needs checking
OK to check the root filesystem (/dev/root) (y/n)?
```

This message is displayed only if the system was not stopped properly, as described in "Stopping the system" (page 52).

Each filesystem generates a similar message. In order to work properly, the operating system requires clean filesystems. If the above message does not appear, your filesystem is clean and ready to use. To clean the filesystem, enter **y** (for "yes") and press the <Return> key. The **fsck(ADM)** utility cleans the filesystem, repairing damaged files or deleting files that cannot be repaired. It reports on its progress as each step is completed. At some point, you may be asked if you wish to salvage a file. Always answer by entering **y** or **n** and pressing the <Return> key. For an explanation of how **fsck** works, refer to "Repairing filesystems with **fsck**" (page 115) for more information.

When cleaning is complete, the system asks you to choose the mode of operation.

Choosing the mode of system operation

You may choose the mode of operation as soon as you see the message:

```
INIT: SINGLE USER MODE
```

```
Type CONTROL-d to continue with normal startup,  
(or give the root password for system maintenance):
```

The system has two modes: normal operation and system maintenance. Normal operation is for ordinary work on the system. This is the mode that allows multiple users to log in and begin work. It is also known as multiuser mode. System maintenance mode is reserved for work to be done by the system administrator, and does not allow multiple users. It is also known as single user mode.

To choose normal operation, press `<Ctrl>d`. The system displays a startup message, and you are prompted to enter the system time (see the next section). Then the system executes commands found in the `/etc/rc` directories, (this includes the `/etc/rc.d` and `/etc/rc2.d` directories, and so forth, referred to collectively as `/etc/rc` scripts) generating startup messages for the various system services, such as the printer or network services. (These scripts are described later in this chapter.) Next, the system displays the “login:” prompt. You can now log in as a normal user, or as the super user.

To choose system maintenance mode, enter the super user password (also called the “root password”) and press `<Return>`.

NOTE The super user (*root*) password is assigned during system installation. If you do not know the *root* password, ask the administrator who installed your system.

The super user prompt “#” is displayed. The commands in the `/etc/rc` scripts are not executed. (Choose system maintenance mode only if you must do system maintenance work that requires all other users to be off the system.) When you log out of system maintenance mode using `<Ctrl>d`, the system automatically enters normal operation.

Entering system maintenance mode by shutting down first

To go from normal operation to system maintenance mode, log in as *root* and give the following command to shut down the system:

```
/etc/shutdown -gn
```

△ **sysadmsh** users select: System ⇔ Terminate

where *n* is the number of minutes until multiuser mode is stopped. After *n* minutes has elapsed, you are asked to confirm your choice and the system is shut down. You are then asked to press any key to reboot. When the system has rebooted, give the *root* password to enter single user mode (see “Choosing the mode of system operation” (page 45) for more information).

NOTE If you attempt to select System ⇔ Terminate from within a window, (as opposed to executing **sysadmsh** from the UNIX command line) it will fail. The easiest way to shut down the system is to log in as *root* and use the **shutdown** command.

Entering system maintenance mode directly

To go from normal operation to system maintenance mode directly, log in as *root* and give the following command:

```
/etc/shutdown -gn su
```

where *n* is the number of minutes until multiuser mode is stopped. The **su** indicates that you want to go directly into single user mode.

NOTE There is no **sysadmsh** equivalent for this command.

Setting the time and date

Once normal operation starts, the system asks for the correct time and date. It displays the current time and date and then the following message:

```
INIT: New run level: 2

Current System Time is Wed Nov 13 08:19:00 PST 1991
Enter new time ([yyymmdd]hhmm):
```

Unless your clock battery is drained or removed, there should be no need to change the date. To leave the time and date unchanged, simply press **<Return>**. If you need to change the time and date, enter the new time and press **<Return>**.

The new values must be entered as two or more consecutive pairs of digits, where the digits may be one or more of the following:

yy (optional)	represents the current year. It may be any two-digit value, from 70 to 99 for the years 1970 to 1999, respectively.
mm (optional)	represents the current month. It may be any two-digit value, from 01 to 12 for January to December, respectively.
dd (optional)	represents the current day. It may be any two-digit value, from 01 to the last day of the month.
hh	represents the current hour. It may be any two-digit value, from 00 to 23. Hours are expressed in military time, where morning hours range from 00 to 11 and evening hours from 12 to 23.
mm	represents the current minutes. It may be any two-digit value, from 00 to 59.

For example, to change the time and date to February 3, 1991 at noon, enter:

9102031200

Press **<Return>**. After accepting the new value, the system then displays the new time and date:

Sun Feb 03 12:00:00 PST 1991

If you enter an incorrect value, the system prompts you to try again. If you do not enter an optional value, the current value for that item remains unchanged. If you type a new value for the year, you must also type values for the month and day. Similarly, if you type a new value for the month, you must type a value for the day.

The time and date display is followed by service startup messages and the "login:" message.

Checking the security databases

Each time your system is rebooted (and after **fsck** is run if your system was brought down unexpectedly) the system automatically checks critical security database files. The messages appear as follows:

```
Checking tcb ...
Checking auth database ...
Checking protected subsystems database ...
Checking ttys database ...
```

This checking is done to avoid problems with gaining access to your system. In the rare case where a file is missing, you are alerted to this fact and asked to restore the file from backups.

When the system is halted suddenly by power or hardware failures, some file-system damage can occur. Such damage can result in the removal of security database files, or can leave these files in an interim state if they were being updated at the time of the system crash. Whenever a reboot occurs, the system runs a series of programs to check the status of the database files. When the system terminates abnormally and is rebooted, this check is performed after `fsck(ADM)` is run on the root filesystem, and before entering multiuser mode.

This check proceeds as follows:

1. The script `/etc/smmck` (system maintenance mode checker) runs the `tcback(ADM)` program to clean up any database files that were left in an interim state while being updated.

When a security database file is updated, the contents of the old file (*file*) is copied or updated to create the new “-t” file (*file-t*). Next, the old file (*file*) is moved to a “-o” file (*file-o*), and the new file (*file-t*) is moved to the original name (*file*). When this process is interrupted, “-o” and “-t” files are left and must be reconciled before the system will function properly. `tcback` first resolves any “-t” and “-o” files left in the `/etc/auth/system`, `/etc/auth/subsystems`, `/etc/files/auth/*` directories, and the `/etc/passwd` and the `/etc/group` files. If there are multiple versions of a file, the extra files must be removed. This is done automatically as follows:

- a. If *file*, *file-o*, and *file-t* exist and *file* is not zero length, then *file-t* and *file-o* are removed.
- b. If *file* and *file-t* exist then *file-t* is removed.
- c. If only *file-t* exists, then it is moved to *file*.
- d. If only *file-o* exists, then it is moved to *file*.

If scenario C occurs, a message similar to the following is displayed:

```
/etc/tcback: file file missing, saved file-t as file
```

This is done because the “-t” file is the modified version of the original file and could have been damaged; it is likely that this file does not contain all the entries of the original. This message is repeated for all files found in that state in the specified directories. (The “-o” files are not suspect because they are the original versions of the files renamed prior to updating.)

WARNING If you do not have backups and the files `/etc/group` and `/etc/passwd` were regenerated by moving the “-t” files, do not restore the original files from your distribution media. The “-t” files will have most (if not all) of your entries, and the distribution versions will not have any.

- Next, `tcback` checks that key system files are present and that they are not of zero length. If a file is missing (or zero length), then a message similar to that shown below is displayed:

```
/etc/tcbck: file file is missing or zero length
```

This process is repeated for each of the following files:

```
/etc/auth/system/default†
/etc/auth/system/files
/etc/auth/system/devassign
/etc/auth/system/authorizet
/tcb/files/auth/r/root†
/etc/group
/etc/passwd†
```

When this process is complete, if any files were missing, or empty “-t” files were substituted for real files, the following message is displayed:

```
/etc/smmck: restore missing files from backup or distribution.
```

NOTE Corrupted files are not detected by `tcback`, but other error messages are displayed; these messages are described in “Security system troubleshooting” in *Performance and Troubleshooting*.

- If critical database files are removed or corrupted (files marked with a dagger (†)), then the system enters maintenance mode automatically without asking for the root password. The messages appear as follows:

```
INIT: SINGLE USER MODE
Security databases are corrupt.
Starting root shell on console to allow repairs.
Entering System Maintenance Mode
```

If no critical database files are missing, you are prompted to choose system maintenance mode or normal operation.

- If you find that files must be restored, your first option is to restore the files from your backups. The second option is to restore the files from the SCO Open Desktop/SCO Open Server distribution media.

For example, if the system reported that the file `/etc/group` was missing and you had a backup of the root filesystem that was created using `sysadmsh(ADM)`, you would use the Backups ⇨ Restore ⇨ Partial selection and restore the file. If backups are unavailable, you will have to use the distribution files.

† critical tcb files

NOTE If you must restore the file */tcb/files/auth/r/root* from your distribution, retrieve volume N2 and insert it into your floppy drive, then enter the following commands:

```
cd /tcb/files/auth/r
mount -r /dev/install /mnt
cp /mnt/tcb/files/auth/r/root .
```

5. Use **custom(ADM)** to restore any missing files. You can restore only one file at a time; do not try to specify more than one file. When you have restored all the necessary files, exit **custom** and press (Ctrl)d. You are then prompted to enter system maintenance mode or normal operation. Press (Ctrl)d again.
6. **tcbck** then removes the files */etc/auth/system/pw_id_map* and */etc/auth/system/gr_id_map* because the modification times of these files are compared with those of */etc/passwd* and */etc/group*, and problems can occur when the system clock is reset. **tcbck** then tries to rebuild the map files using **cps(ADM)**. If this fails then either the File Control database (*/etc/auth/system/files*) is missing, or the the File Control database entry for *" / "* is missing, or there are syntax errors in */etc/passwd* or */etc/group*.
7. After the system enters multiuser mode ("**INIT: New run level: 2**" is displayed) and you are prompted to set the system clock, */etc/authckrc* is reinvoked. If any missing files are found, warnings similar to the ones shown previously are displayed, followed by the message shown below:

```
/etc/tcbck: file file is missing or zero length
/etc/authckrc: Log in on the OVERRIDE tty and restore
the missing files from a backup or the distribution disks.
```

This means that you missed some files earlier. These files will have to be replaced when the system comes up in multiuser mode and you are allowed to log in. Write down the names of the missing files at this stage.

8. Next, the following message is displayed:

```
Checking auth database ...
```

The **authck(ADM)** program is run to make certain that all users listed in */etc/passwd* have Protected Password database entries. If any are missing, they are created as needed.

9. Next, the following message is displayed:

```
Checking protected subsystem database ...
```

The **authck(ADM)** program is run. The Protected Subsystem database files are checked to ensure that they correctly reflect the subsystem authorization entries in the Protected Password database. Each name listed in each subsystem file is verified against the Protected Password entry with the same name, so that no authorization is inconsistent between the files. Also, each Protected Password entry is scanned to verify that all the

privileges listed are reflected in the Protected Subsystem database. If any inconsistencies are found, you are asked if you want them fixed automatically:

```
There are discrepancies between the databases.
Fix them (Y or N)?
```

The error messages are found in the **authck(ADM)** manual page.

NOTE If the system is autobooting ("AUTOBOOT=YES" appears in */etc/default/boot*), then **authck(ADM)** is called noninteractively. Warnings are displayed about inconsistencies found but **authck** is not given the opportunity to fix them. The transition to the multiuser operation then proceeds as normal.

You can also run **authck** manually; see the manual page for syntax.

10. Next, you see the following message:

```
Checking ttys database ...
```

ttypd(ADM) is run to ensure that all ttys in */etc/inittab* have entries in the Terminal Control database (*/etc/auth/system/ttys*).

11. The system is now up and ready for logins. If any files were reported missing, you must now log in on the override terminal to restore them, following the same procedure outlined earlier. By default, the override terminal is defined as *tty01*, also known as the first multiscreen. If you removed the default entry in */etc/default/login*, you will have to shut the system off, reboot and enter single-user mode, and restore the files that way. When you log in on the override tty, the following message is displayed:

```
The security databases are corrupt.
However, root login at terminal tty01 is allowed.
```

Logging in as the super user

Many system maintenance tasks, when performed during normal operation, require you to log in as the super user. For example, you must be logged in as the super user to stop the system.

To log in as the super user, you must know the super user password. You also need to see the "login:" message on your terminal's screen. If you do not see this message, press **(Ctrl)d** until it appears.

3. Starting and stopping the system

NOTE The super user (*root*) password is assigned during system installation. If you do not know the *root* password, ask the administrator who installed your system.

To log in as the super user, follow these steps:

1. When you see the login message, enter the super user login name:

```
login: root
```

Now press the `<Return>` key. The system prompts you for the super user password.

2. Enter the super user password and press the `<Return>` key. The system does not display the password as you enter it, so enter each keystroke carefully.

The system opens the super user account and displays the message of the day and the super user prompt `"#"`.

Take special care when you are logged in as the super user. In particular, you should be careful when deleting or modifying files or directories. This is important because the super user has unlimited access to all files; it is possible to remove or modify a file that is vital to the system. Avoid using wildcard designators in filenames and keep track of your current working directory.

You can leave the super user account at any time by pressing `(Ctrl)d`.

Stopping the system

Stopping a UNIX system requires more than just turning off the computer. You must prepare the system for stopping by using the `shutdown(ADM)` command.

Using the `shutdown` command

The `shutdown` command is the normal way to stop the system and should be used whenever the system is in normal operation mode. It warns other users that the system is about to be stopped and gives them an opportunity to finish their work. (The warning message that `shutdown` displays at all terminals can be customized; see the `shutdown(ADM)` manual page for details.) If desired, the system administrator can also use the `wall(ADM)` command to send a message about the impending shutdown prior to running the actual `shutdown` command.

To stop the system with the **shutdown(ADM)** command, follow these steps:

1. Log in as the super user. Refer to "Logging in as the super user" (page 51) for more information. The system opens the super user account and displays the message of the day and the super user prompt.
2. Enter the following command and press (Return):

```
/etc/shutdown -gn
```

Δ sysadmsh users select: System ⇨ Terminate

where *n* is the number of minutes before the shutdown is to take place. The system displays a warning message at each terminal, asking logged-in users to finish their work and to log out. As soon as all users are logged out or the specified time has elapsed, the system closes all accounts and displays the following message:

```
** Safe to Power Off **
-or-
** Press Any Key to Reboot **
```

WARNING If you attempt to select System ⇨ Terminate from within a window, (as opposed to executing **sysadmsh** from the UNIX command line) it will fail. The easiest way to shut down the system is to log in as *root* and use the **shutdown** command.

3. Turn off the computer or press any key to reboot the system.

Understanding the boot display information

At boot time, a table of hardware information is always displayed after the copyright information. This table represents your hardware configuration as recognized by the operating system. Example 3-1 is an annotated version of the boot screen as it appears on a sample machine.

Example 3-1 Sample boot display

device	address	vector	dma	comment
fpv	-	35	-	type=80387
floppy	0x03F2-0x03F7	06	2	unit=0 type=96ds15
serial	0x02F8-0x02FF	03	-	unit=1 type=Standard nports=1
parallel	0x0378-0x037A	07	-	unit=0
console	-	-	-	unit=ega type=0 12 screens=68k
disk	0x01F0-0x01F7	36	-	type=W0 unit=0 cyls=791 hds=16 secs=48

3. Starting and stopping the system

This information is explained in the following key:

device	the name of the hardware
address	address in hexadecimal
vector	interrupt vector
dma	direct memory access channel
comment	other details about the hardware
fpu	floating-point unit present, specifically the Intel 80387 and 80486 chips
floppy	high density 5.25" floppy drive (type=96ds15)
serial	this is COM1; COM1 has one port (nports=1, no multiport card is installed)
parallel	this is parallel port lp0 (unit=0)
console	the console has an EGA video adaptor (unit=ega) compatible with (type 0) the IBM EGA design
disk	Western Digital st506 controller number 0 (W0), hard drive 0 (unit 0), as well as the number of cylinders, heads, and sectors

The `hwconfig(C)` utility displays or accesses this information at any time, using the configuration information stored in the file `/usr/adm/hwconfig`. Refer to the `hwconfig(C)` manual page for more information.

In addition, the `eisa(ADM)` utility can be used to list the cards installed in EISA machines, and the `slot(ADM)` utility used for MCA machines.

Changing the boot process

Each time the computer is started, the system runs the `boot` program. Unless you give different instructions at the prompt, `boot` loads the default kernel program using the configuration values specified in the file `/etc/default/boot` on the default root filesystem. You can change the current boot process with your response to the prompt, or you can edit the `/etc/default/boot` file to change the default configuration values for future boot operations. You can also set certain options in `/etc/default/boot` to allow the system to boot automatically. The `boot(HW)` manual page describes the default boot options.

Booting an old kernel

If you have recently relinked the kernel and it fails to boot properly or causes other problems, you can enter `unix.old` at the boot prompt and load the previous kernel.

Changing the default bootstring (DEFBOOTSTR)

To change which program is loaded by default when you enter only a **<Return>** at the boot prompt, modify the default bootstring set with the **DEFBOOTSTR** option in */etc/default/boot*. For example, the following setting in */etc/default/boot* causes the **boot** program to load the kernel from a hard disk by default.

```
DEFBOOTSTR=hd(40)unix
```

The first two letters of the argument specify the device ("**hd**" for hard disk or "**fd**" for floppy disk). The number in parentheses is the minor device number (40 for the root filesystem on the hard disk). Following the parentheses is the pathname of the program to be loaded.

The *Hardware Configuration Guide* and the **boot(HW)** manual page describe certain keywords or "bootstrings" that you can add to the **boot** command line to load special drivers at boot time. You can also add these to the end of the default bootstring set with **DEFBOOTSTR**. For example, here is the bootstring for a Wangtek cartridge tape:

```
DEFBOOTSTR=hd(40)unix ct=wangtek(0x338,5,1)
```

Booting automatically

The settings of the **AUTOBOOT**, **TIMEOUT**, and **PANICBOOT** options in */etc/default/boot* control if, and when, automatic booting occurs.

If **AUTOBOOT=NO**, the **boot** program waits indefinitely for a response to the prompt. If **AUTOBOOT=YES**, the system boots automatically if no response is given at the prompt after a certain amount of time. The amount of time **boot** waits for a response before booting automatically is 60 seconds by default, but you can set the number of seconds to wait with the **TIMEOUT** option. If a timeout occurs, **boot** behaves as though you entered a **<Return>** to the boot prompt, performing the default boot process using the configuration values specified in */etc/default/boot*.

You can set the **PANICBOOT** option to **YES** or **NO** to indicate whether or not the system should reboot after a panic (a panic always causes the system to halt). Refer to the **autoboot(ADM)** manual page for more information.

Chapter 4

Administering user accounts

User accounts help the system administrator keep track of the people using the system and control their access to system resources. Accounts also help organize user files and protect them from access by other users. Each account has a unique “login name” and “password” with which the user enters the system, and a “home directory” where the user works. In addition, the system has certain defaults that define how long a user password should last, whether users are allowed to choose their own passwords, and how many unsuccessful login attempts should be allowed before locking a user out.

It is the system administrator’s job to create accounts for all users on the system, and to maintain these accounts by changing user passwords, login groups, and other account parameters when necessary.

There is the option to have several system administrators, each charged with limited control over certain aspects of system operation, or a single *root* user controlling all administration operations (*root* exists with full super user powers by default).

This chapter covers the following topics:

- Account information: for those who are familiar with UNIX systems, this section explains how database files such as */etc/passwd* and */etc/shadow* are used and how they coexist with the additional C2-related databases.
- Account management: how to add, alter, and remove (or retire) user accounts, plus create user groups.
- Default account configuration: how to configure system default login and password parameters, and if desired, change the security scheme.

It is important to examine the default account restrictions soon after creating user accounts. Refer to "Default account configuration" (page 81) for a summary. You should determine if these defaults are appropriate to the needs of your system.

How account information is stored

One of the principal differences between the various UNIX systems is in how account information is stored. This affects the migration of accounts across different UNIX systems, and governs how programs access this data. The files fall into two categories: UNIX system files (those defined in the System V Interface Definition) and the Trusted Computing Base (TCB) files that extend System V security. The following files are supported and maintained by the system to ensure compatibility with other UNIX systems (files marked with an asterisk are TCB files):

- */etc/passwd* file. This publicly readable file is present on most UNIX systems and contains both account data (user ID number, login shell, and so forth) and (on some systems) an encrypted account password. Password aging information is also supported. It can be edited by experienced administrators, but *sysadmsh* is the preferred method for adding and maintaining user accounts. If your system is configured with the Improved or High security defaults and you edit */etc/passwd* manually, you must run the *authck(ADM)* command with the *-p* and *-y* options to update the Protected Password database. In the lower modes, you can edit */etc/passwd* without running *authck*.
- */etc/shadow* file. This file is readable only by *root*. It contains the encrypted password otherwise found in the */etc/passwd* file. (When the */etc/shadow* exists, */etc/passwd* contains an "x" in the password field.) This file is manipulated via the *pwconv(ADM)* and *pwunconv(ADM)* utilities. This file exists by default in all security defaults except Low, where it still can be created using *pwconv*.
- */etc/default/passwd* and */etc/default/login* files. These contain default account information that, in many cases, is duplicated in the Protected Password and System Defaults database.
- * Protected Password database (*/tcb/files/auth/[a-z]/username*). This database implements the requirements for the C2 level of trust as defined by the *Trusted Computing System Evaluation Criteria* (TCSEC). It contains the encrypted password of the user as well as their privileges (authorizations), password parameters, and other detailed information. The format of this file is described in the *authcap(F)* manual page.

- * System Defaults database (*/etc/auth/system/default*) file. This contains the information used for creating default accounts; the contents of this file are determined by the security defaults selected (Low, Traditional, Improved, or High). The format of this file is described in the *authcap(F)* manual page.

All database files are updated automatically when a change is made via *sysadmsh*.

In the event of a discrepancy between these files, either the UNIX System V files or the TCB databases are used as the master to bring them into agreement. In the Low and Traditional security defaults the UNIX System V files are the master*. You can also configure which set of files is used as the master set; this is described in the next section.

Configuring database recovery and precedence

To define the recovery scheme to be used when a database failure occurs, make the following *sysadmsh* selection:

Accounts ⇄ Defaults ⇄ System

The following screen is displayed:

System

Allow duplicated account information to be inconsistent

/ Monday August 31, 1992 1:06

System wide security

Fail if password and Protected Password entries differ: Yes No

Protected Password database entry is the master : Yes [No]

Length of cleartext password (units of 8 characters) : [1]

* Refer to "Default account configuration" (page 81) for information on security defaults.

The following parameters define database redundancy and precedence:

Fail if password and Protected Password entries differ

If Yes, this enforces consistency in account information; an error message is generated when an inconsistency occurs and no action is taken. If No, inconsistent data is permitted (and automatically corrected) between the different database files. Under the Improved and High security defaults, users are locked out of the system when a discrepancy occurs until the administrator logs in on the override terminal to fix the problem.

Protected Password database entry is the master

If Yes, this ensures that the trusted database files are used as the master files and information is written to the traditional UNIX system data files only for consistency — it is never relied upon for data used by the system. If No, the traditional System V data files (*/etc/passwd* and so forth) are used as the master files and the trusted databases (Protected Password and System Default) are written to only for consistency and are not relied upon for data used by the system. Under the Low and Traditional security defaults, the System V files are used as the master.

Length of cleartext password

Because the password encryption schemes on most other XENIX and UNIX systems support a maximum password length of eight characters (pre-encryption), this field provides compatibility by controlling the maximum length of passwords. The term “cleartext” refers to an unencrypted password, so password length in this instance is expressed in units of eight characters. Under the Low and Traditional security defaults, this field has a value of 1, which maintains compatibility with other UNIX systems (meaning that the */etc/passwd* file can be imported to other systems).

Account management

This section explains how to create and manage user accounts.

Adding a user

You can add a user account to the system with **sysadmsh(ADM)**, which creates a new entry in the Accounts database. The database contains information about the new user (such as login name and initial password) that the system uses to let the user log in and begin work. **sysadmsh** also creates a home directory for the user, a mailbox for use with the **mail** command, and an initialization file (for example, *.profile* for the Bourne and Korn shells or *.login* for C-shell) containing UNIX system commands that are executed when the user logs in.

NOTE Although `sysadmsh` is the preferred method for adding and maintaining user accounts, experienced administrators can edit the `/etc/passwd` file directly. If your system is configured with the Improved or High security defaults and you edit `/etc/passwd` manually, you must run the `authck(ADM)` command with the `-p` and `-y` options to update the Protected Password database. In the lower modes, you can edit `/etc/passwd` without running `authck`.

To create a user account, make the following `sysadmsh` selection:

Accounts ⇨ User ⇨ Create

The following screen is displayed:

```

Name of new user (once set, this cannot be changed)
/ Monday August 31, 1992 1:06

Make a new user account

Username : [ ]
Comment  : [ ]
Modify defaults? Yes [ No ]
  
```

Follow these steps to add a user:

1. Fill in the "Username" and, if desired, "Comment" fields.
2. If you wish to alter the defaults, select Yes and define the fields as shown in "Modifying account creation defaults" (page 64) . Fill in each field as necessary; press `(F3)` to choose from point-and-pick lists. When you press `(Return)`, the field is filled in with the value you selected.
3. When you exit the form, a window pops up to confirm your additions. If confirmed, a series of creation messages are displayed that look like this:

```

Created home directory: pathname
Created shell file: filename
Greetings mail sent to user: name
  
```

This indicates that all the necessary files and directories were created. (This default information is taken from the directory */usr/lib/mkuser*.)

NOTE If you see a message similar to the following:

```
useshell: Warning
useshell: File Control database inconsistency
useshell: Script path /usr/lib/mkuser/mkuser.init may be compromised
          - run integrity(ADM)
```

this means there may be a problem with the permissions of the database file indicated (including any part of the path). If your system is configured with the Low or Traditional security defaults, this message can be ignored. For Improved or High defaults, this can be remedied by using the **fixmog**(ADM) utility to correct permissions on the system (use **fixmog -i** to fix any problems interactively). We recommend that you run the **integrity**(ADM) utility first to get a report of all incorrect permissions on the system, but note that **integrity** does not make changes.

4. Finally, **sysadmsh** prompts you as to whether an initial password should be created. The following menu is displayed:

The screenshot shows a terminal window with the following content:

```
Assign the new account a password (the user then can log in)
/ Monday August 31, 1992 1:06

Make a new user account

Username : [ sample ]

Assign an initial password

Until a password is assigned to user : name
no-one may log in as that account.

Assign first password : Now Later Blank Remove

Force change at first login : [ Yes ] No
```

The possible selections for "Assign first password" are as follows:

Now	assigns the new account a password
Later	does not assign the new account a password (the user cannot log in)

- Blank** assigns the new account an empty password (the user is asked to set a password at first login)
- Remove** assigns the new account no password at all (the user can log in without a password)

The "Force change at first login" field governs whether users must change their passwords the first time they log in. Note that setting this to Yes negates the effect of Remove; that is, the user is required to set a password when logging in.

5. If you selected to generate a password for the new user, you see the following prompt:

```
Setting password for user user
Password change is forced for user

      Choose password

You can choose whether you pick your own password,
or have the system create one for you.

      1. Pick your own password
      2. Pronounceable password will be generated for you

Enter choice (default is 1):
```

6. If you select 1 and you are operating under the Improved or High security defaults, you see the following prompts:

```
Please enter new password (at least 5 characters):
Please choose a password which contains a mixture of lower-
and upper-case letters, digits (0 - 9), and non-alphanumeric
characters (e.g., !, #, @, ;, %, or /.)

Please do NOT choose a password that is an English word,
or which is the name of a person, place, or thing, or which
contains the string "SCO", "XENIX", or "UNIX" (in either case).
Re-enter password:
```

Note that the password is not displayed on the screen as you enter it. You are asked to enter another password if the one you entered is unacceptable.

7. If you select 1 and you are operating under the Low or Traditional security defaults, you see the following prompts:

```
Please enter new password (at least 1 character):
New password:
```

You can also simply press **(Return)** to establish a null password, where the user presses **(Return)** when prompted for the password.

8. If you select 2, the following is displayed:

```
Generating random pronounceable password for user.
The password, along with the hyphenated version, is shown.
Hit <RETURN> or <ENTER> until you like the choice.
When you have chosen the password you want, type it in.
Note: Type your interrupt character or 'quit' to abort at any time.
Password:xxxxxxx Hyphenation:xx-xx-xx Enter password:
```

The generated password is displayed with a hyphenated version. The hyphenation separates the password into pronounceable syllables and is designed to help you commit the password to memory.

9. Give the new password to the user. If you selected to force a password change, the user is required to change it immediately after logging in for the first time.

The new account is usable and is maintained according to the default security parameters unless you have set specific values for the user.

Modifying account creation defaults

For most users, the defaults should prove sufficient. If you selected to "Modify Defaults," you see the following form:

Create

Use the system default login group

/ Monday August 31, 1992 1:06

Make a new user account

New user account parameters

Login group : Specify **Default** of
Value, (F3) for list :
Groups : [...]

Login shell : Specify [Default] of sh
Value, (F3) for list :

Home directory : Specify [Default] of [/usr/name]
Value, (F3) for list : [
[Create home] Do not create Populate existing

User ID number : [Specify] [Default] of [] value : []

Type of user : Specify [Default] of individual
Value, (F3) for list :

Account that may su(C) to this user :

The cursor is initially positioned on the "Login group" field. Some of the fields displayed can only be modified at creation time, in the Modify mode. These fields are informational only and their values cannot be changed.

The fields are as follows:

Login group

This is the group associated with this account when the user logs in. This field can be changed, but must not be empty. It becomes the group field for this user in */etc/passwd*. Pressing function key (F3) provides a point-and-pick list of all currently existing groups.

Groups

These are the groups this user is a member of. If you enter the name of a group that does not exist, it is automatically created as described in "Adding or changing groups" (page 76).

Login shell

This is the shell the user will use. (The default is defined in */etc/default/authsh*.) If a full pathname is supplied (as in */bin/sh*), the shell described by that pathname is simply used as the user's login shell. However, if the shell specified is not a pathname (as in *sh*), it is assumed to be the name of a "predefined shell," a shell defined in a subdirectory of */usr/lib/mkuser*. Choosing a predefined shell causes appropriate shell-related files (for example, *.profile* for *sh*) to be copied into the user's home directory when the account is created. (Refer to "Adding login shells and configuration files" (page 94) for more information on creating your predefined shells.)

Home directory

This defines where the user's files will reside. The default directory option is highlighted. Press (Return) to get the default location. The home directory options are as follows:

Create	creates a new directory for this user
Do not create	does not create a directory for this user
Populate existing	uses the existing directory specified

If you wish to have users share a single home directory, refer to "Sharing home directories" (page 66) for instructions.

User ID number

This is the user identification (ID) number. Once set, a user's identity should not be changed as this would violate accountability.

Type of user

By default this is "individual," and you need not change it. Pseudo-users are anonymous accounts like *sysinfo* and *uucp*. Each pseudo-user has an "accountable user," who is considered responsible for that account. (For example, *root*, an individual, is defined as the accountable user for all pseudo-user accounts.) Several user types are provided for customers who need to make distinctions between accounts that can be used by different people.

Account that may su(C) to this user

This is the user responsible for this account. This field is only useful with the High security defaults, where strict accountability is maintained and use of su(C) is restricted. Under other defaults, this field can be ignored. This field can be changed if and only if the user is not an individual. For individual users this field is empty, but for non-individuals it must not be empty. It must contain the user name of another account. For example, all pseudo-user accounts shipped with the system are owned by *root*, and defined as an “individual” account. This can be set up so that every account can be traced to a real person. Press (F3) for a point-and-pick list of all users on the system.

Sharing home directories

You can set up user accounts to share the same login directory. To do this, create the directory normally during the adding of a new user. You should then exit *sysadmsh* and enter the following commands (replacing *homedir* with the real directory name):

```
cd homedir
chmod 775 .
chown auth .
```

In addition, enter one of the following commands according to the login shell used for the account:

Bourne or Korn shell:

```
chmod 660 .profile
chmod 660 .kshrc      (Korn shell only)
```

C-shell:

```
chmod 660 .login .cshrc
```

This ensures that the members of the same login group can share this directory. Note that if you assign a different login group to a user, that user cannot share the directory. To add users that share the directory, be sure to select “Modify defaults,” specify the directory, and “Populate existing” directory when creating subsequent users. Answer *n* when asked to overwrite the “dot” files such as *.profile* and so forth.

Adding administrative users

In addition to the standard Identity information, users who act as administrators for printers, accounts, and so forth, can be assigned the responsibility by selecting Privileges. Under lower security defaults, most subsystem authorizations are already assigned by default, so this distinction is meaningless. (The *auth* authorization gives *root* powers and is only assigned to *root* by default, regardless of the security defaults.)

WARNING The **auth** subsystem authorization should only be assigned to persons entrusted with account administration. This is because it permits the user to make changes to any account, including *root*.

Subsystems are discussed in “Changing default authorizations” (page 90) and assigning authorizations is discussed in “Changing user authorizations” (page 74).

Altering the defaults displayed for user accounts

You can alter the default selections that appear in the user creation menu by editing the file */etc/default/authsh*. The following defaults can be redefined:

- login group
- groups
- login shell
- home directory
- range of user IDs
- type of user

The fields in */etc/default/authsh* are explained in the **authsh(ADM)** manual page.

Removing or retiring a user account

When operating under High (C2) security defaults, a user is never removed from the system. Once assigned, a user ID (UID) is never reused. Instead, a user account is “retired,” or removed from service. Under other defaults, a user can be completely removed from the system. To retire a user account, make the following **sysadmsh** selection:

Accounts ⇄ User ⇄ Retire

When running with lower security defaults, you can completely remove accounts using the **rmuser(ADM)** utility. To remove a user, enter the following command, substituting the name of the user account for *username*:

rmuser username

Retiring or removing a user account does not remove the user’s files; the system administrator must do this manually.

Unretiring a user account

If you are running under lower security defaults and you wish to “unretire” or reactivate a user’s account, you can do so with the **unretire(ADM)** command. To reactivate a retired user, enter the following command, substituting the name of the user account for *username*:

unretire username

Locking or unlocking a user account

The system administrator can lock an account to prevent its use. In addition, under the Improved and High security defaults, an account is locked automatically if certain login parameters have been exceeded (see “Default account configuration” (page 81) for more information).

NOTE Under the Low and Traditional security defaults, accounts and terminals are only locked by the administrator; there are almost no limits on the number of unsuccessful login attempts, and password expiration is less strict.

Once a user or terminal is locked, only an administrator can unlock the user account or terminal (terminal locks are discussed in the next section). To lock or unlock an account, make the following `sysadmsh` selection:

Accounts ⇨ User ⇨ Examine:Logins

The colon indicates that you must fill in a field (in this case the user name) before choosing the Logins selection. A form similar to the following is displayed:

Logins

Use the system default limit on unsuccessful login attempts

Monday August 31, 1992 1:06

View/Modify an existing user's account

Login history and locks

Username	: sample	
Last login attempt Location Date/time		
successful :	tty01	Mon 7 May 1991 03:22:06 AM
unsuccessful :	tty2b	Thu 3 May 1991 08:22:06 AM
Last logout	: tty2b	Mon 7 May 1991 07:19:24 AM
Number of unsuccessful login attempts since last successful login : 1		
Maximum number of unsuccessful attempts before account is locked		
	Specify Default of 6	Value :
Account locked : NO LOCKS		
Lock status	[No change]	Apply administrative lock Clear all locks

Move down to the “Lock status” field and toggle it to “Apply administrative lock” or “Clear all locks” as desired.

Locking or unlocking a terminal

To lock and unlock a terminal, respectively, use the following `sysadmsh` selections:

Accounts ⇌ Terminals ⇌ Lock
Accounts ⇌ Terminals ⇌ Unlock

When the prompt appears for the terminal, enter the name, for example: `tty01`. When a terminal is locked, the following message is displayed when an attempt is made to log in:

Terminal is disabled -- see Authentication Administrator

Changing a user's login group

To change a user's login group, make the following `sysadmsh` selection:

Accounts ⇌ User ⇌ Examine:Identity

The colon indicates that you must fill in a field (in this case the user name) before making the Identity selection. A form similar to the following is displayed:

Identity

Group associate with this account when the user logs in ({F3} for list)

/ Monday August 31, 1992 1:06

View/Modify an existing user's account

Identity

Username	: sample		
User ID	: 246	Type of user	: individual
		Account that may su(C) to this user	: NONE
Login group	: [pub]		
Groups	: [...]		
Login shell	: [/bin/sh]		
Home directory	: [Keep]	Edit	Create Move Restore
path	: /usr/sample		
Comment	: []		
Priority	: Specify [Default] of 0 Value:		

Modify the "Login group" field as desired.

Changing a home directory

To change a user's home directory, including moving the user's files, make the following `sysadmsh` selection:

Accounts ⇌ User ⇌ Examine:Identity

The colon indicates that you must fill in a field (in this case the user name) before making the Identity selection. A form similar to the following is displayed:

Identity

Group associated with this account when the user logs in (<F3> for list)

Monday August 31, 1992 1:06

View/Modify an existing user's account

Identity

Username	: sample		
User ID	: 246	Type of user	: individual
	Account that may su(C) to this user : NONE		
Login group	: [pub]		
Groups	: [...]		
Login shell	: [/bin/sh]		
Home directory	: [Keep]	Edit	Create Move Restore
path	: /usr/sample		
Comment	: {		
Priority	: Specify [Default] of 0 Value:		

After moving down to the "Home directory" field, you can select to do the following:

- Keep** makes no change to the main directory for this user
- Edit** changes this user's main directory path, but does not move any files
- Create** creates the main directory for this user
- Move** renames this user's main directory, moving all files in the old to the new
- Restore** changes the path back to its previous value (no files are moved)

Changing a user password or password parameters

An administrator can change a user's password at any time. Password generation parameters can also be changed on an individual basis, just as they can be system-wide. This governs how a user's password is changed. To do this, make the following `sysadmsh` selection:

Accounts ⇨ User ⇨ Examine:Password

The colon indicates that you must fill in a field (in this case the user name) before making the Password selection. The following form is displayed:

```

Use the system default maximum password length for this user

/ Monday August 31, 1992 1:06

View/Modify an existing user's account
Password selection

Username : sample

Password required : Yes No Default of Yes

User can choose own : Yes No (Default) of Yes

User can run generator : Yes No (Default) of Yes
Maximum generated length : Specify (Default) of 10 Value :

Checked for obviousness : Yes No (Default) of No

Current password status : [ Keep ] Change Disable Remove
Change password at login : Yes [ No ]
  
```

The Password selection parameters are similar to the system-wide parameters described in "Changing default password restrictions" (page 87). The following parameters define the per-user password restrictions:

Password required

If Yes, the user cannot log in without a password; if No, the user can log in without a password.

User can choose own

This parameter determines whether or not users can choose their own passwords. If this parameter is set to Yes, users can pick their passwords. If that parameter is set to No, the system must generate a password for that user, according to the random password generation procedures.

NOTE If “User can choose own” and “User can run generator” are both No, users cannot change their own passwords.

User can run generator

This parameter enables the user to run the password generator. Note that this does not allow the user to choose a password, but merely to generate a new random password.

Maximum generated password length

This parameter is the maximum length of a password generated by the system for this user. The maximum is 80 characters.

Checked for obviousness

This parameter is only valid with Improved or High security defaults, or when `GOODPW=YES` appears in `/etc/default/passwd`. This controls whether the system should run simple or complex triviality checks on a new password. Triviality checks increase the time required to change a password substantially. Refer to “Customizing password checking with `goodpw(ADM)`” (page 89) for more information.

Current password status

You can choose one of the following:

- Keep does not change password.
- Change invokes the password change procedure described in “Adding a user” (page 60).
- Disable disables the password, which effectively locks the user out.
- Remove removes the password, enabling the user to log in without a password.

Change password at login

This option allows you to force users to change their password the next time they log in.

Altering user password expiration parameters

It is sometimes useful to define expiration parameters for a user that differ from the system defaults. To do this, make the following `sysadmsh` selection:

Accounts ⇨ User ⇨ Examine:Expiration

The colon indicates that you must fill in a field (in this case the user name) before choosing the Expiration selection. A form similar to the following is displayed:

```

Use the system's default minimum password lifetime

/ Monday August 31, 1992 1:06

View/Modify an existing user's account
Password life and death

Username      : sample

Last password change  Date/time           Days ago
successful : Wed Feb 22 09:27:29 1991      42
unsuccessful : Fri Feb 24 10:55:03 1991      40

Minimum number of days between password changes :
Specify (Default) of 14 Value :
Maximum number of days before password must be changed :
Specify (Default) of 42 Value :
Maximum number of days before account is locked for not changing password :
Specify (Default) of 365 Value :

```

The user parameter descriptions are similar to the system-wide parameters described in “Changing default password restrictions” (page 87). The descriptions differ, but the parameters are the same. They control the time that a password is valid, and the procedures for changing the password once it becomes invalid. The lifetime of a password is divided into three intervals:

- The password is valid.
- The password has expired; the user can still log in and change it (if authorized to do so).
- The password is dead; the user is locked out and the administrator must unlock the account and the user change the password.

NOTE The Low and Traditional security defaults use password restrictions that are very lenient; passwords do not expire, accounts are not locked, and there is no minimum interval between password changes.

The system also stores a minimum time between password changes. This prevents users from changing their passwords when they expire and then immediately changing them back to one they remember. A user's password cannot be changed until the minimum time has been exceeded. The password lifetime scheme is implemented as follows:

Minimum number of days between password changes

This defines the number of days users must wait before they can change their password.

Maximum number of days before password must be changed

This defines the length of time a given password is valid.

Maximum number of days before account is locked for not...

This defines the interval between the last password change and when the password dies.

The default account initialization files (*.cshrc*, *.profile*, *.kshrc*, and so forth) automatically execute the **prwarn(C)** utility at login time to warn users about impending password expiration.

Changing user authorizations

Authorizations define what users are allowed to do on the system. Authorizations are discussed in detail in "Changing default authorizations" (page 90). Authorizations are also used to assign administrative roles; for more information see the "Maintaining system security" chapter of *Open Server System Administrator's Guide* in the Open Desktop Supplementary Documentation.

NOTE The Low and Traditional security defaults assign most authorizations to users by default and it should not be necessary to change them. With the High (C2) security defaults, few authorizations are assigned by default; they are intended for users entrusted with administration of a subsystem. Only users entrusted with system administration should be assigned the **auth** subsystem authorization.

To assign a new authorization to a user, make the following **sysadmsh** selection:

Accounts ⇌ User ⇌ Examine:Privileges

The colon indicates that you must fill in a field (in this case the user name) before choosing the Privileges selection.

The following form is displayed:

```

Auths

Use default kernel authorizations

/ Monday August 31, 1992 1:06

View/modify an existing user's account
Authorizations

Username : sample

Kernel : Specify Default authorizations: [... ]

Subsystem : Specify [Default] authorizations: [... ]

When specifying authorizations
(F3) will list those which may
be selected.
  
```

You can select Specify and press (F3) to open a window that lists the available authorizations. Use the arrow keys to move up and down the list. You can select multiple authorizations by pressing the space bar; each item selected is marked with an asterisk (*).

NOTE If you switch from defaults to specified, the default values are eliminated for that user; only those authorizations you specify are in effect.

Changing user audit parameters

You can define audit parameters for individual users just as with the system-wide parameters. Any settings defined for a user override the system defaults.

NOTE It is not necessary to modify audit parameters if you do not intend to use auditing.

To define or change audit parameters, make the following `sysadmsh` selection:

Accounts ⇨ User ⇨ Examine:Audit

The colon indicates that you must fill in a field (in this case the user name) before choosing the Audit selection.

4. Administering user accounts

A form similar to the following is displayed:

The screenshot shows a terminal window titled 'Audit'. The main text reads 'System startups (boots) and shutdowns'. Below this is a status bar showing the path '/' and the date/time 'Monday August 31, 1992 1:06'. A sub-header 'View/Modify an existing user's account' is followed by the title 'Audited Events'. The form contains a text field for 'Username: sample' and a table of 20 audit events, each with a default setting.

Audited Events	
Username: sample	
A. Startup/Shutdown	[Default]
B. Login/Logoff	[Default]
C. Process Create/Delete	[Default]
D. Make Object Available	[Default]
E. Map Object to Subject	[Default]
F. Object Modification	[Default]
G. Make Object Unavailable	[Default]
H. Object Creation	[Default]
I. Object Deletion	[Default]
J. DAC Changes	[Default]
K. DAC Denials	[Default]
L. Admin/Operator Actions	[Default]
M. Insufficient Authorization	[Default]
N. Resource Denials	[Default]
O. IPC Functions	[Default]
P. Process Modifications	[Default]
Q. Audit Subsystem Events	[Default]
R. Database Events	[Default]
S. Subsystem Events	[Default]
T. Use of Authorization	[Default]

Refer to the “Using the audit subsystem” chapter in the *Open Server System Administrator's Guide* in the Open Desktop Supplementary Documentation for a detailed description of audit events.

There are three possible settings for each event:

- Default use the system-wide account defaults
- Always always audit this event (overrides system-wide default)
- Never never audit this event (overrides system-wide default)

You can press <F3> to select from a list of these settings or fill them in manually. Abbreviations are recognized (for example, “n”, “nev”, and “N” all mean Never). To execute the form, press <Ctrl>x. (If you fill in the last field on a form, it is automatically executed.)

Adding or changing groups

To add a group, simply enter a new group name while creating or altering a user account. You are prompted that the group does not exist and asked to confirm that you wish to create the new group.

You can also edit the file */etc/group* and add or modify the contents as desired.

WARNING Do not change the GIDs of any default system groups.

Supplemental groups

On most System V implementations, a user can be a member of several groups, but can only have a single effective group ID (GID), which limits membership to one group at a time. The effective GID (and group membership) is changed using the `newgrp(C)` command. The list of groups is located in `/etc/group`.

The supplemental groups feature allows an additional set of groups to be defined, permitting simultaneous membership in multiple groups. This provides additional access permissions on top of those provided by the user and group IDs.

The `.suppgroups` file, which is recognized by programs that establish identity (`login`, `su` and `cron`), specifies which groups (of which the user is a member) are placed in the user's supplemental group list when identity is established. The `.suppgroups` file contains one group name per line. If a `.suppgroups` file does not exist, the initial supplemental group list is filled with the login group ID followed by the groups the user is listed as being a member of in `/etc/group`. They are listed in the same order as in `/etc/group`. The `.suppgroups` file does not need to be readable or writable by the user.

The `sg(C)` command is used to manipulate a user's supplemental group set. This command does not establish identity so it does not read the `.suppgroups` file.

Changing the maximum number of supplemental groups

By default, the maximum size of a supplemental group set is eight, meaning that a user can only be in eight groups at one time. This number is controlled by the `NGROUPS` tunable kernel parameter. This value can be changed by invoking the following `sysadmsh` selection:

System ⇨ Configure ⇨ Kernel ⇨ Parameters

and selecting category 3, "Files, Inodes, and Filesystems," and changing the value of `NGROUPS`. The kernel must then be relinked and booted for the new value to take effect. You can use this selection to relink the kernel:

System ⇨ Configure ⇨ Kernel ⇨ Rebuild

Allowing users to execute super user commands

It is possible to assign users the capability of executing *root*-only commands without giving them complete *root* powers. This is done using the `asroot(ADM)` utility to create a new authorization associated with the

command you wish to assign. You can then add this authorization to any user. You can also assign the **root** subsystem authorization to permit the use of all commands configured with **asroot**. The procedure for setting up a super-user command for this usage is described in detail in the **asroot(ADM)** manual page.

NOTE The **asroot** utility requires the user's password to be re-entered when the system is configured with the High security defaults.

Accessing other accounts with su(C)

The **su(C)** utility (for super user) can be used to switch over to another account temporarily. It is primarily used to access the **root** account, when it is executed without an argument. Otherwise, it is used in the following form:

su username

su prompts for the account password, and if it is correct, a Bourne shell is started under the other account. Transitions with **su** do not affect the login user ID (LUID), so login and audit records remain accurate. If a dash (-) is included in the command (**su -**), the environment for that user is executed (including login shell, home directory, and so forth). To exit the shell, enter **exit** or press **(Ctrl)d** and you are returned to your own account.

Users can **su** to an account they are defined as responsible for in the "Account that may **su(C)** to this user" field of the Accounts ⇨ User ⇨ Examine:Identity **sysadmsh** selection. To access the **root** account (or any other account they are not responsible for), however, the user must have the **su** authorization. This can be assigned using the Accounts ⇨ User ⇨ Examine:Privileges selection. Refer to "Changing user authorizations" (page 74) for more information.

NOTE The Low, Traditional, and Improved security defaults assign the **su** authorization by default and users can **su** to any account if they know the password. Under the High security defaults, the **su** authorization is not assigned.

Moving user accounts to another filesystem or directory

If you want to move your user accounts to a different location (such as a new filesystem or directory), you must start by ensuring that new accounts are placed in the new location. Here we use a hypothetical filesystem called **/x** as an example. The **sysadmsh(ADM)** Accounts option (used to create new user accounts) reads the default location for user accounts from the **/etc/default/authsh** and **/usr/lib/mkuser/homepaths** files.

Edit the files `/etc/default/authsh` and `/usr/lib/mkuser/homepaths`. Change the entries that read `"/usr"` to `"/x"`. This establishes `/x` as the location for new user accounts.

Whenever you run the `sysadmsh` Accounts selection to add a new user, that user account is placed in `/x`. (Make certain that the filesystem containing the user accounts is mounted before you create a new user, or the user's home directory will not be accessible when the filesystem is mounted.)

If there are existing user accounts that you wish to relocate, you should first make a backup floppy or tape of all home directories you plan to move. (This protects your users from possible loss of data.) Next, change the home directory for each user by calling up their account information with the following `sysadmsh(ADM)` selection:

Accounts ⇄ User ⇄ Examine:Identity

Move to the "Home Directory" field and select the Move option. Follow the prompts. Do this for every user whose home directory has changed.

Migrating user accounts to non-SCO-based UNIX systems

You can migrate user accounts to other XENIX or UNIX systems by copying the `/etc/passwd` (and `/etc/shadow`, if applicable) to the target system. If the target system does not use `/etc/shadow`, you should run the `pwunconv` utility to consolidate the information into `/etc/passwd` before copying the file to the target system.

NOTE If you need to migrate accounts to another SCO-based UNIX system, use the `ap(ADM)` utility described in the next section.

Password compatibility

The password encryption scheme used by the system maintains compatibility with other XENIX and UNIX system implementations, while providing the ability to create passwords with more than eight significant characters. However, if you are using the Improved or High security defaults, passwords of up to 80 characters are allowed, which cannot be imported to other systems. The `sysadmsh` selection Accounts ⇄ Defaults ⇄ System includes the field "Length of cleartext password". A value of 1 (instead of 10 as used in the Improved or High defaults) allows the encryption mechanism to ignore characters following the first eight, thus allowing complete compatibility with other systems. Refer to "Configuring database recovery and precedence" (page 59) for more information.

Migrating user accounts to SCO-based UNIX systems

You can easily duplicate user accounts on other SCO-based UNIX systems (as with a network) using the `ap(ADM)` utility. This utility creates a profile containing all account data for one or more users.

WARNING The `ap(ADM)` utility does not create profiles that are portable to non-SCO-based UNIX systems. If you need to migrate accounts from a non-SCO-based UNIX system or SCO XENIX system, use the `addxusers(ADM)` utility described in the next section.

Account information is gathered from the `/etc/passwd` file and the Protected Password database. Irrelevant information about the user (including unsuccessful logins, unsuccessful password changes, and the location and time of last login) is not included in the profile.

To create a profile and install it on another machine, do the following:

1. Log in as `root` and enter the following command on the machine where the accounts reside:
`ap -d -v usernames > profile.acct`
usernames is the list of one or more user names.
2. Log in as `root` and move the *profile.acct* file to the target machine (use `tar`, or `cp` if your machine is on a network).
3. Enter the following command:
`ap -r -f profile.acct usernames`

The new account information is in place and ready for use.

Migrating user accounts from SCO XENIX or non-SCO-based UNIX systems

To copy accounts from other non-SCO-based UNIX systems (or SCO XENIX systems) to your system, use the `addxusers(ADM)` utility. `addxusers` accepts an edited `/etc/passwd` file as input and makes the necessary database modifications for use on your system. Refer to the `addxusers(ADM)` manual page for more information.

WARNING Most XENIX and UNIX systems only use the first eight characters for encryption. This can cause unexpected results when moving an encrypted password string from one of these systems to an SCO-based UNIX system. If a password longer than eight characters has been used, such as "narcissus", only the first eight characters ("narcissu") should be entered on a SCO-based UNIX system.

Default account configuration

This section explains how to alter the system security defaults, which include default password schemes, subsystem authorizations, and number of login attempts permitted for users.

The system includes four sets of defaults that define the security scheme for user accounts:

- Low
- Traditional
- Improved
- High

The “High” defaults are designed to meet the requirements set forth by the Department of Defense’s *Trusted Computer System Evaluation Criteria* (also known as TCSEC or the *Orange Book*). Table 4-1 lists the defaults used by each set. The Low and Traditional defaults use values consistent with non-trusted UNIX systems. Selecting the security defaults is described later.

NOTE The pre-defined defaults can be customized as desired; the four default sets are provided for ease of configurability.

Table 4-1 System default security parameters

Security parameters	Low	Security defaults		High
		Traditional	Improved	
Passwords				
Minimum days between changes	0	0	0	14
Expiration time (days)	infinite	infinite	42	42
Lifetime (days)	infinite	infinite	365	90
User can choose own	yes	yes	yes	no
User can run generator	yes	yes	yes	yes
Maximum generated length	8	8	10	10
Minimum length	1	3	5	8
Checked for obviousness	no	UNIX	yes ¹	yes ²
Password required to login	no	no	yes	yes
Single user password required	yes	yes	yes	yes
Logins				
Maximum unsuccessful attempts (account/terminal)	infinite	99	5/9	3/5
Delay between login attempts (secs)	0	1	2	2
Time to complete login (secs)	60	60	60	60
Authorizations				
Subsystem	audittrail, backup, lp, mem, queryspace, shutdown, su, terminal	audittrail, mem, printqueue, queryspace, terminal, su	audittrail, queryspace, printqueue, su	queryspace
Kernel	chmodsugid, chown, execsuid, suspendaudit	chmodsugid, chown, execsuid	chmodsugid, chown, execsuid	chown, execsuid
Default umask ³	022	022	027	077
C2 Features				
LUID enforcement ⁴	no	no	no	yes
STOPIO on devices ⁴	no	yes	yes	yes
SUID/SGID clear on write ⁴	no	yes	yes	yes
Users can be deleted ⁵	yes	yes	no	no
Database corruption ⁶	recover	recover	lockout	lockout
Database precedence ⁷	UNIX	UNIX	TCB	TCB
Other				
Users can schedule jobs	allow	allow	deny	deny
Home directory mode	755	755	750	700
Dialup printers allowed	yes	yes	no	no
Hushlogin allowed ⁸	yes	yes	yes	no
Password for asroot(ADM)	no	no	no	yes
Significant characters in passwords	8	8	80	80
su(C) use logged	no	yes	yes	yes
/etc/shadow present	no	yes	yes	yes

Notes:

1. Simple checks are made, such as disallowing user names, machine names, etc...
2. Thorough checks are made, including disallowing words that appear in the on-line dictionary.
3. These are located in */etc/profile* and */etc/cshrc*. A **umask** of 077 results in the creation of files that are readable only by the owner.
4. These features are controlled by kernel parameters. See the section on security kernel parameters in *Performance and Troubleshooting* for more information.
5. A requirement central to C2 is that a user ID (UID) cannot be reused. This means that user accounts cannot be reused or reactivated after retirement. With the lower security defaults, user accounts can be removed rather than retired and user IDs can be altered or reused.
6. On a system that conforms to C2 requirements, users are locked out of a system when a security database becomes corrupted. This ensures that the system does not operate in a potentially non-secure state. In the lower defaults, the system attempts to correct inconsistencies automatically and displays a warning rather than locking out users.
7. Two sets of account databases are maintained: UNIX System V and trusted computing base (TCB) files. One set is used as a master when a discrepancy occurs. This is described in "Configuring database recovery and precedence" (page 59).
8. This feature allows the suppression of login messages. See **login(M)** for information.

Changing the security defaults

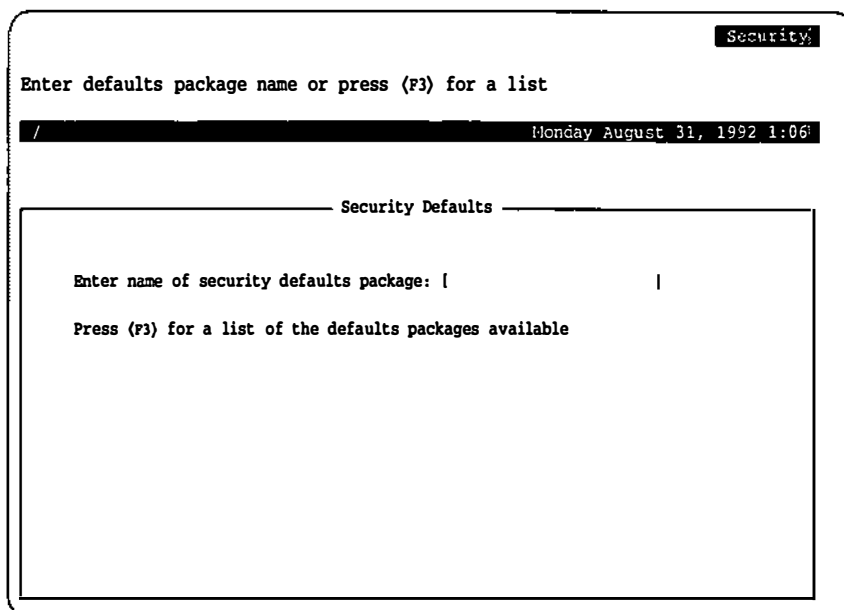
You were given the choice of security defaults at installation time. It is possible to later select another set of defaults. Should you wish to change the security defaults of your system, make the following **sysadmsh** selection:

System ⇨ Configure ⇨ Security

This allows you to select an alternate set of defaults (see Table 4-1).

NOTE After having selected lower security defaults, it is possible to select the Improved or High defaults, although this does not mean that your system automatically conforms to the requirements of a C2 system. (By definition, a C2 system must adhere to the requirements from initial installation.)

When you make the Security selection, the following is displayed:



The screenshot shows a terminal window titled "Security". The prompt is "Enter defaults package name or press (F3) for a list". Below the prompt is a status bar showing a slash "/" and the date/time "Monday August 31, 1992 1:06". The main content area is titled "Security Defaults" and contains the prompt "Enter name of security defaults package: |" with a cursor. Below this is the instruction "Press (F3) for a list of the defaults packages available".

You may reconfigure your system security to suit your own requirements. Four sets of preconfigured security defaults are available:

High	recommended for systems containing confidential information and accessed by many users. Passwords are strictly controlled and assigned to users; they cannot choose their own. User accounts cannot be removed or reactivated. All C2 features are engaged and database corruption results in a lockout of all users until the administrator fixes the problem.
Improved	recommended for systems accessed by groups of users who can share information. Password expiration is more lenient and users can choose their own passwords. LUIDs are not enforced, and user accounts can be removed or reactivated as desired. Database corruption results in system lockout.
Traditional	provided for compatibility with existing UNIX systems. Passwords do not expire and standard System V password checking is used. Passwords are not required. Database corruption is handled transparently.
Low	recommended only for systems not publicly accessible with a small number of cooperating users. No C2 features are engaged and no password checking is done. The <i>/etc/shadow</i> does not exist by default.

The system is designed to meet the requirements for the C2 level of trust describing the protection given to prevent unauthorized access to a system and its data. The High and Improved defaults are suitable for C2 systems.

Changing system account parameters dynamically

In addition to selecting a package of default parameters, the following system-wide account parameters can be customized individually:

- authorizations
- password
- logins

The system-wide security parameters control the way that users log in and, once they establish a session, the terminal and authorization environment that the system presents to them. Each parameter that you can change from the `sysadmsh` interface is discussed here. Other parameters that affect system operation are addressed later.

You should use the system-wide functions to define your own default system behavior. Then use the user-specific functions to adjust that behavior for any user having different requirements. As you might expect, the user-specific entries override the system defaults for any given user.

Changing default login restrictions

Most of the configurable security parameters deal with the way the system creates a login session. These include login particulars and the way that passwords are generated and enforced. The login parameters enforce the account and terminal-locking features. When users log in, they must give a login name and password. In addition, the user has a limited number of tries to log in. There is a limit on the number of times an unsuccessful login attempt can occur before either the account or the terminal are locked. If either count is exceeded, the user or the terminal is locked against future login. This feature guards against penetration attempts by restricting the number of times a malicious user (or computer programmed by a malicious user) can try to break into the system.

NOTE The Low and Traditional security defaults use login restrictions that are very lenient; there are almost no limits on the number of unsuccessful login attempts and little delay between attempts.

To access the login restriction parameters, make the following `sysadmsh` selection:

Accounts ⇄ Defaults ⇄ Logins

The following form is displayed:

Login:

Allowed consecutive failed login attempts before account is locked

/ Monday August 31, 1992 1:06

Login Restrictions

Maximum number of unsuccessful attempts before locking ...

... user account: [5]

... terminal : [5]

Delay (in seconds) between login attempts on a terminal : [2]

Time (in seconds) to complete successful login : [40]

CPU scheduling priority after successful login : [0]

The parameters are described as follows:

Maximum number of unsuccessful attempts before locking

This is the system default number of unsuccessful attempts allowed for users and terminals. If a particular user or terminal needs either a more restrictive or more permissive number, the user's account can be modified or the terminal's configuration can be changed to override the system default. To modify an individual terminal, use the Accounts ⇨ Terminal ⇨ Examine selection.

Delay (in seconds) between login attempts on a terminal

This parameter controls the amount of time that must pass between unsuccessful login attempts. To further reduce the possibility of penetration, the system can delay between login attempts to increase the amount of time it takes to try to log into the system repeatedly. You can increase this parameter to control the cycle time of the "login:" prompt. By combining this parameter with the user and terminal unsuccessful attempt parameters, you can frustrate attempts to try passwords repeatedly on certain (or a combination of) terminal lines.

Time (in seconds) to complete successful login

This parameter determines how much time users have to enter their name and password before the login attempt is terminated.

CPU scheduling priority after successful login

This sets the `nice(C)` value associated with this user's processes. This allows you to set a higher or lower CPU priority for a user. See the `nice(C)` manual page for details.

Changing default password restrictions

Given that you can control the number of attempts an intruder can try to guess a password, the remaining task is to control the complexity of the password itself. To access the password restriction parameters, make the following `sysadmsh` selection:

Accounts ⇄ Defaults ⇄ Password

The following screen is displayed:

Monday August 31, 1992 1:06

Minimum number of days which must elapse between password changes

/

Password Selection

Minimum days between changes	: [14]	
Expiration time (days)	: [42]	
Lifetime (days)	: [365]	
User can choose own	: [Yes]	No
User can run generator	: [Yes]	No
Maximum generated password length	: [10]	
Checked for obviousness	: Yes	[No]
Password required to login	: [Yes]	No
Single user password required	: [Yes]	No

The types of password checking the system does is controlled by the parameters set on this screen. The parameters control the time that a password is valid, and the procedures for changing the password once it becomes invalid. A password is valid until it "expires" or "dies." An expired password can be changed by whomever is authorized to change passwords for the account. A password expires when its expiration time is reached. The expiration time can be set from `sysadmsh` on a system-wide or a per-user basis, and it is expressed in number of days from the time that the password was last changed. A dead password causes the user account to be locked. Only the administrator can unlock the user's account, which is then treated as an account with an expired password. The password must still be changed before the user can log in again.

NOTE The Low and Traditional defaults use password restrictions that are very lenient; passwords do not expire, accounts are not locked, and there is no minimum interval between password changes. In addition, there is no password checking with the Low defaults; Traditional defaults use the customary UNIX system password checks.

The system also stores a minimum time between password changes. This prevents users from changing their passwords when they expire and then immediately changing them back to one they remember. A user's password cannot be changed until the minimum time has been exceeded. This parameter may also be set on a per-user or system-wide basis.

The following parameters define the password restrictions:

Minimum days between password changes

The number of days a user must wait between password changes.

Expiration time (days)

This defines the length of time a given password is valid. The default account initialization files (*.cshrc*, *.profile*, *.kshrc*, and so forth) automatically execute the *prwarn(C)* utility at login time to warn users about impending password expiration.

Lifetime (days)

This defines the interval between last password change and when the password dies.

User can choose own

This parameter determines whether or not users can choose their own passwords. You can choose to have the system generate passwords automatically for users. This guards against users picking "obvious" passwords that a knowledgeable intruder could guess given some personal facts about the user. Most UNIX systems allow users to pick their passwords. If this parameter is set to Yes, then rules consistent with non-trusted UNIX systems are in effect, allowing users to pick their passwords. If that parameter is set to No, the system must generate passwords for that user, according to the random password generator.

User can run generator

This parameter enables users to run the password generator. Note that this does not allow users to choose a password, merely generate a new random password.

Maximum generated password length

This defines the maximum length of a password generated by the system. The maximum is 80 characters.

Checked for obviousness

This parameter is only valid with the Improved or High security defaults, or when "GOODPW=YES" appears in */etc/default/passwd*. This controls whether the system should run simple or complex triviality checks on passwords. These checks assure that the password does not appear in the online dictionary, along with the other checks described in *goodpw(ADM)*. Setting this parameter to Yes ensures the failure of some penetrations based on dictionary checking, but this can be controlled more effectively through the limits on account and terminal logins. Triviality checks increase the time required to change a password substantially. Refer to "Customizing password checking with *goodpw(ADM)*" (this page) for more information.

Password required to login

If No, user accounts can be without passwords; if Yes, a password must exist for the account.

Single user password required

This governs whether a password is required to bring the system up in single-user (maintenance) mode.

When an account is locked by the system, only *root* or the accounts administrator can unlock it. The password must be changed at that time. Refer to "Changing a user password or password parameters" (page 71) for information on overriding these parameters for any user.

Customizing password checking with *goodpw(ADM)*

The *goodpw(ADM)* utility allows you to customize password checking. The file */etc/default/goodpw* contains the password control settings. These settings allow you to specify if passwords are checked against dictionary words, word rotations, user, group, and system names.

NOTE Password checking can also be set by editing the */etc/default/passwd* and changing the value of **GOODPW** as follows:

YES	use <i>goodpw</i>
NO	use standard UNIX system checking
NONE	perform no password checking

In addition, the directory */usr/lib/goodpw/checks* allows you to further customize password requirements according to the type of password:

- user
- filsys (filesystem)
- group
- modem

Each type has a file (*secure* or *weak*) that is used depending on whether the "Check for obviousness" field is set to Yes (*secure*) or No (*weak*).

You can also define regular expressions (character combinations and arrangements) that all passwords must match, or must not match, with the files */usr/lib/goodpw/match* and */usr/lib/goodpw/reject*, respectively. For more details, refer to the *goodpw(ADM)* manual page.

Changing default authorizations

There are two types of authorizations: *Subsystem authorizations* are associated with users and allow the user to execute trusted utilities. *Kernel authorizations* are associated with processes and allow a process to perform certain actions. Each user session has a set of kernel authorizations and subsystem authorizations.

WARNING The *auth* subsystem authorization should only be assigned to persons entrusted with account administration. Never assign *auth* by default because it permits users to make changes to any account, including *root*. Refer to the sections that follow for more information.

To access the authorization parameters, make the following *sysadmsh* selection:

Accounts ⇌ Defaults ⇌ Authorizations

The following screen is displayed:

The screenshot shows a terminal window titled "Authorization". The header "Privileges enforced by the system" is at the top left, and a status bar at the top right shows the date and time: "Monday August 31, 1992 1:06". Below the header, the word "Authorizations" is centered. Underneath, it says "System default authorizations ({F3} for list)". There are two labels: "Kernel:" and "Subsystem: [...]". Under "Kernel:", a list box contains the following items: "chmodsugid", "chown", "execsuid", and "suspendaudit".

You can use the **(Tab)** key to move between kernel and subsystem authorizations. Use the **(F3)** key to get a pop-up window that lists each set of authorizations. Detailed descriptions of authorizations appear in the sections that follow.

Subsystem authorizations

Subsystem authorizations were designed to implement administrative roles rather than using a single *root* user to administer the system. Under the Low and Traditional security defaults, most subsystem authorizations (except **auth**) are assigned to users by default.

If you intend to operate a system that conforms to C2 requirements, you should grant subsystem authorizations based on the notion of “least-privilege”, where users are assigned subsystem authorizations based on their responsibilities. For example, the backups administrator is granted the **backup** authorization and the printer administrator is granted **lp** authorization. (Only *root* should have all authorizations.) Under this scheme, general users should be assigned as few subsystem authorizations as possible. Secondary authorizations are provided to grant limited capabilities from a subsystem.

The subsystem authorizations are listed in Table 4-2.

Table 4-2 Subsystem authorizations

Authorization	Subsystem	Powers
mem	memory	access to system data tables, listing all processes on the system
terminal	terminal	unrestricted use of the write(C) command
lp	line printer	administer printers
backup	backups	perform backups
auth	accounts	administer accounts: adding users, changing passwords, and so on
audit	audit	audit administrator: run system audits and generate reports
cron	job scheduling	control use of cron(C) , at(C) , and batch(C) commands
root	asroot(ADM)	allows use of any command found in <i>/tcb/files/rootcmds</i> (see “Allowing users to execute super user commands” (page 77) for more information)
sysadmin	—	not implemented
passwd	passwords	ability to change user passwords

Secondary authorizations allow limited access by users to resources that would otherwise be tightly controlled (for example, without the **printqueue** authorization, users would only be able to see their own jobs when they use the **lpstat** command). They are useful when running the Improved or High security defaults to provide behavior that is more consistent with other UNIX operating systems. See Table 4-3.

Table 4-3 Secondary authorizations

Secondary authorization	Subsystem	Description
audittrail	audit	ability to generate personal audit reports on one's own activities
queryspace	backup	use of df(C) command to query disk space
printqueue	lp	view all jobs in queue using lpstat(C)
printerstat	lp	use of printer enable/disable commands
su	auth	access to the root (super user) account and other accounts. (Access still requires password.)
shutdown	asroot(ADM)	ability to use the shutdown(ADM) command.

NOTE When the primary authorization for a subsystem is granted, the secondary authorizations for that subsystem are also granted. (For example, the **lp** authorization carries the **printqueue** and **printerstat** authorizations.)

Kernel authorizations

The kernel authorizations govern the power that user processes have to execute specific operating system services. For example, the ability to change ownership of a file is governed by the **chown** authorization. (The **chown** authorization allows the use of the **chown(S)** system call which enables **chown(C)** to work.) The default kernel authorizations are used whenever a user's kernel authorizations are unspecified. Thus, users that need more authorization can have user-specific entries that grant them those authorizations, while normal users can have their authorizations set to the system-wide defaults. See Table 4-4.

NOTE Restricted **chown** is required for NIST FIPS 151-1 conformance. The **chown** authorization should not be assigned to users if you wish to conform to NIST FIPS 151-1 requirements.

Table 4-4 Kernel authorizations

Authorization	Action
configaudit	configure audit subsystem parameters
writeaudit	write audit records to the audit trail
execsuid	ability to run set-UID programs
chmodsugid	ability to set the set-UID and set-GID bit on files
chown	ability to change the owner of an object
suspendaudit	suspend operating system auditing of the process

Under the Low and Traditional security defaults, most kernel authorizations are assigned by default and should not require modification. Under the High security defaults, **chmodsugid** is not assigned by default. Most users require only **execsuid** kernel authorizations to perform routine tasks. If the user needs to create files with the SUID or SGID bits, they must have the **chmodsugid** authorization. To change ownership of a file (give it away), the **chown** authorization is required. If a user does not have this authorization, ownership of files can only be changed by *root*. The audit kernel authorizations (**configaudit**, **writeaudit**, and **suspendaudit**) should never be assigned to anyone other than the audit administrator. They are intended for use by a program designed to run as a trusted application.

Kernel authorizations and administrative users

If you are operating with the High and Improved security defaults, you must assign certain kernel authorizations along with subsystem authorizations. Although most of these are already assigned by default, they are listed in Table 4-5 in case you modify the defaults. One exception is the audit subsystem, which requires the addition of the **configaudit** and **suspendaudit** authorizations. These authorizations should never be assigned by default, or to ordinary users.

NOTE Under the Low and Traditional security defaults, most kernel authorizations are already assigned by default.

Table 4-5 Subsystem kernel authorization requirements

Subsystem authorization	Kernel authorization required
audit	configaudit, execsuid, writeaudit
auth	execsuid, chown
backup	execsuid
lp	chown
cron	chmodsugid, chown, execsuid
sysadmin	chmodsugid, chown, execsuid

Adding login shells and configuration files

It is possible to add files for additional login shells that can be selected for users when they are created using `sysadmsh`. For example, `csch(C)` has prototype `.login` and `.cshrc` files that are installed in a user's directory when `csch` is selected as that user's login shell. Each shell has a directory of these prototype files in `/usr/lib/mkuser`. You can examine the existing files and follow their example. Make sure the permissions and ownership are consistent with the other files. When you install the files, the new shell is a valid selection in the "Login shell" field of the user Identity form.

Chapter 5

Managing filesystems

This chapter describes one of the most important responsibilities of a system administrator: creating and maintaining filesystems. General maintenance activities are described, such as strategies for maintaining free space. The concept of “filesystem integrity” is introduced, with a description of how the operating system repairs damaged filesystems. Filesystems reside on hard disks, CD-ROMs, and floppy disks. Adding hard disks and CD-ROM drives is described in the *Hardware Configuration Guide*. After adding a new device to the system, filesystems residing on them must then be added to the system using `mkdev fs` as described in this chapter.

What is a filesystem?

A filesystem is a distinct division of the operating system, consisting of files, directories, and the information needed to locate and access them. A filesystem can be thought of as a structure upon which directories and files are constructed.

Each UNIX system has at least one filesystem on the primary hard disk. This filesystem is called the “root filesystem” and is represented by the symbol `/`. The root filesystem contains the programs and directories that comprise the operating system. On small hard disks, the root filesystem also includes all the user directories. The primary hard disk can also be divided into more than one filesystem. This is described in the *Installation and Upgrade Guide*. One of the most common divisions is the `/u` filesystem, used to isolate user accounts from the root filesystem.

A UNIX system can also have other filesystems that contain special directories and application programs. Dividing the primary hard disk into multiple filesystems protects the data and makes maintenance easier. Adding still more filesystems by installing other hard disks expands the system storage space.

New filesystems can be specifically created by the system administrator, then “attached” (mounted) and “detached” (unmounted) when needed, in the same way that a floppy disk is accessed. The process of adding a hard disk is described in the *Hardware Configuration Guide*. Adding filesystems after a hard disk is configured is described in the following section.

Adding new filesystem(s)

After configuring a new hard disk, you must then add the new filesystem(s) to the system. To do this, follow these steps:

1. Enter the following command:

```
mkdev fs
```

```
Δ sysadmsh users select: Filesystems ⇨ Add
```

2. You see the following:

```
This program performs maintenance tasks required to add or delete
an existing filesystem. Would you like to:
```

1. Add a new filesystem to system.
2. Remove a filesystem.

```
Select an option or enter q to quit:
```

Select 1.

3. You are next prompted for the device name:

```
Enter a device name and press <Return> or q to quit:
```

Enter the full pathname of the device from */dev*. The device name is derived from the name you gave the filesystem during the *divvy* phase. For example, to add a filesystem called *u*, you enter “*/dev/u*”.

4. You are now prompted to provide the name of the mount point to be used:

```
Enter a directory name and press <Return> or q to quit:
```

This directory is where the filesystem is mounted. For example, a filesystem called *u* is mounted on the directory */u*.

5. The following is displayed:

```
Reserving slots in lost+found directory ...

When entering multiuser mode:
  1. Always mount filesystem
  2. Never mount filesystem
  3. Prompt before mounting filesystem

Select an option:
```

If you want the filesystem mounted automatically at system startup, enter 1. If you wish it mounted only by the request of the system administrator, select 2. If you select 3, the system prompts you at system startup whether or not you want the filesystem mounted.

6. You are then asked whether or not you want to permit users to mount filesystems:

```
Do you want to allow users to mount this file system? (y/n)
```

You must respond "y" so that the system backup program can mount and unmount the filesystem as necessary.

7. The following messages are displayed when the process is complete.

```
Updating system files ...
Filesystem has been successfully added.
```

8. Next, you should mount the `/x` filesystem using the following command:

```
mount /dev/x /x
```

Δ sysadmsh users select: Filesystems ⇌ Mount

Mounting and unmounting a filesystem

The **mount(ADM)** command attaches a filesystem. For example, to mount or unmount `/dev/u` on `/u`, you would use the following two commands, respectively:

```
mount /dev/u /u
```

Δ sysadmsh users select: Filesystems ⇌ Mount

```
umount /u
```

Δ sysadmsh users select: Filesystems ⇌ Unmount

Only the super user can use the **mount** and **umount** commands.

NOTE Files in a filesystem are not accessible unless the filesystem is mounted. If files are copied to or created in the mount point directory while the filesystem is unmounted, those files will appear to be in that filesystem when they are not. When the filesystem is mounted, these files seem to “disappear” when the filesystem is mounted over them.

Permitting users to mount filesystems

Only the super user can use the **mount** command. However, the super user can set up parameters to define which filesystems can be mounted by users with the **mnt(C)** command. These parameters may include an access password, if desired.

Each filesystem must have an entry in the file */etc/default/filesys*. Example 5-1 contains a sample set of entries.

Example 5-1 Sample */etc/default/filesys* file

```
bdev=/dev/root cdev=/dev/rroot mountdir=/ \
desc="The Root Filesystem" rcmount=no mount=no

bdev=/dev/u cdev=/dev/ru mountdir=/u rcmount=yes \
fsckflags=-y desc="The User Filesystem"

bdev=/dev/x cdev=/dev/rx mountdir=/x mount=yes \
rcmount=yes fsckflags=-y desc="The Extra Filesystem"
```

The sample entries determine the behavior shown in Table 5-1.

Table 5-1 Filesystem mount specifications

Filesystem	When mounted	Can user mount?
root	boot time	no
/u	multiuser	no
/x	anytime	yes

If you wish to make any non-root filesystem mountable by users, simply add “**mount=yes**” to the entry for the given filesystem. In addition, when the **mnt** command is invoked without an argument (no filesystem name), the program checks all non-root filesystems to see if they can be mounted and, if so, mounts them. The option “**mount=prompt**” asks the user if they want to mount each filesystem where a mount is permitted.

For more information on the **mnt** command, including a complete list of options, refer to the **mnt(C)** manual page.

Configurable filesystem features

There are certain filesystem features that are configurable, including:

- Group ID of newly created files: you can specify whether a new file has the group affiliation of the parent directory or that of the creating user.
- Filename truncation: you can specify what happens when an attempt is made to create a filename longer than the system limit.

This section explains these features.

Setting directory SGID bit

By default, the GID (group identifier) of a newly created file is set to the GID of the creating process or user. This behavior can be changed by setting the SGID bit on a directory. Setting the SGID bit on a directory results in a new file having the GID of that directory. To set the SGID bit on a directory, enter the following command, where *directory* is the directory name:

```
chmod g+s directory
```

To remove the bit, replace the “+” with a “-” in the above command.

Setting filename truncation

By default, attempts to create filenames longer than the system limit result in the error message “Filename too long.” This can be changed so that long filenames are silently truncated to the system limit. The system limit depends on the filesystem type. XENIX, S51K, and AFS filesystems have a limit of 14 characters. The EAFS (default) filesystem has a limit of 255 characters. The default behavior is mandated by POSIX FIPS requirements and is controlled by the ETRUNC kernel parameter. This parameter can be changed by invoking the `sysadmsh` selection System ⇨ Configure ⇨ Kernel ⇨ Parameters and selecting category 3: “Files, Inodes and Filesystems,” and changing the value of ETRUNC to 1. The kernel must then be relinked and booted for the new behavior to take effect. Use the `sysadmsh` selection System ⇨ Configure ⇨ Kernel ⇨ Rebuild to relink the kernel.

Maintaining free space in filesystems

An important task of the system administrator is filesystem maintenance, which includes keeping the system running smoothly, keeping the filesystems clean, and ensuring adequate space for all users. To maintain the filesystems, the system administrator must monitor the free space in each filesystem, and take corrective action whenever the free space gets too low.

This section explains the filesystem maintenance commands. These commands report how much space is used, locate seldom-used files, and remove or repair damaged files.

A UNIX system operates best when at least 15% of the space in each filesystem is free. In any system, the amount of free space depends on the size of the disk containing the filesystem and the number of files on the disk. Because every disk has a fixed amount of space, it is important to control the number of files stored on the disk.

If a filesystem has less than 15% free space, system operation usually becomes sluggish. If no free space is available, the system stops any attempts to write to the filesystem. This means that the user's normal work on the computer (creating new files and expanding existing ones) stops.

The only remedy for a filesystem that has less than 15% free space is to delete one or more files from the filesystem. The following sections describe strategies for keeping free space available.

Strategies for maintaining free space

The system administrator should regularly check the amount of free space on all mounted filesystems and remind users to keep their directories free of unused files. You can remind users by including a reminder in the */etc/motd* (message of the day) file.

In addition, the `cleantmp(ADM)` command is run by the system to clean the */tmp* directory. You can edit the file */etc/default/cleantmp* to define which, and how often, key directories (*/tmp* and */usr/tmp* by default) are cleaned of files. See the `cleantmp(ADM)` man page for details.

If the amount of free space slips below 15%, the system administrator should follow these steps:

1. Send a system-wide message asking users to remove unused files.
2. Locate exceptionally large directories and files, and send mail to the owners asking them to remove unnecessary files.
3. Locate and remove temporary files and files named *core*.
4. Clear the contents of system log files.
5. Reduce disk fragmentation by making a complete backup of the filesystem, removing all the files, and then restoring them from the backup.
6. If the system is chronically short of free space, it may be necessary to create and mount an additional filesystem.
7. `compress(C)` large infrequently used files.

Displaying free space

You can find out how much free space exists in a particular filesystem with the `df` (for "disk free") command. This command displays the number of "blocks" available on the specific filesystem. A block is 512 characters (or bytes) of data.

The `df` command has the form:

`df specialfile`

△ `sysadmsh` users select: System ⇄ Report ⇄ Disk

where *specialfile* can be the name of a UNIX system special file corresponding to the disk drive containing the filesystem. If you do not give a special filename, then the free space of all normally mounted filesystems is given.

For example, to display the free space of the root filesystem `/dev/root`, enter:

`df /dev/root`

The command displays the special filename and the number of free blocks.

You can find the percentage of free space to total space on your system with the command:

`df -v`

Sending a system-wide message

If free space is low, you can send a message to all users on the system with the `wall` (for "write to all") command. This command copies the messages you enter at your terminal to the terminals of all users currently logged in.

To send a message, enter:

wall

Enter the message, pressing <Return> to start a new line if necessary. After you have entered the message, press <Ctrl>d. The command displays the message on all terminals in the system.

Displaying disk usage

You can display the number of blocks used within a directory by using the **du** command. This command can be used to find excessively large directories and files.

The **du(C)** command has the form:

du *directory*

The optional *directory* must be the name of a directory in a mounted filesystem. If you do not give a directory name, the command displays the number of blocks in the current directory.

For example, to display the number of blocks used in the directory */u/johnd*, enter:

du /u/johnd

The command displays the name of each directory in the */u/johnd* directory and the number of blocks used.

Use the **-a** or **-f** options to display files.

Displaying blocks by owner

You can display a list of users and the number of blocks they own by using the **quot** (for "quota") command. The command has the form:

quot *specialfile*

The *specialfile* must be the name of the special file corresponding to the filesystem.

For example, to display the owners of files in the filesystem on */dev/u*, enter:

quot /dev/u

The command displays the users who have files in the filesystem and the numbers of blocks in these files.

Locating files

You can locate all files with a specified name, permissions setting, size, type, owner, or last access or modification date using the **find** command. This command is useful for locating seldom-used or excessively large files, or files owned by a particular user.

The **find** command has the form:

find *pathname option*

The *pathname* is the pathname of the directory that you want to search. The **find** command searches recursively, downward through all the directories under the named directory, for files that match the criteria specified by *option*. Some options also indicate certain actions for **find** to take on located files. See the **find(C)** manual page for complete details.

Table 5-2 describes some of the options to **find** that are useful for system administrators.

Table 5-2 Useful find options

Option	Description
-atime +number	locates files that have not been accessed for the specified <i>number</i> of days.
-exec cmd	locates files that match the specified criteria and executes <i>cmd</i> on those files. The command argument (<i>{}</i>) is replaced by the current pathnames of the files that find located. An escaped semicolon (<i>\;</i>) must follow the <i>cmd {}</i> construction.
-group gname	searches for files that belong to the group <i>gname</i> . If <i>gname</i> is numeric and does not appear in <i>/etc/group</i> , it is interpreted as a group ID.
-name file	searches for files with the specified name.
-ok cmd	works like -exec except that it displays the generated command line in prompt form; <i>cmd</i> is executed only if the user enters <i>y</i> at the prompt.
-perm onum	locates all files with permissions that exactly match <i>onum</i> (the octal number used with chmod(C)).
-print	displays the locations of any files that find locates.
-size +number	searches for files larger than the specified number of blocks (512 bytes per block).

(Continued on next page)

Table 5-2 Useful find options
(Continued)

Option	Description
-type <i>x</i>	locates files of a specific type; for example, type d for directory or f for a file. See find(C) for an explanation of the different types.
-user <i>uname</i>	locates all files that belong to the user <i>uname</i> . If <i>uname</i> is numeric and does not appear in <i>/etc/passwd</i> , it is interpreted as a login ID.

NOTE If you do not include the **-print** option, **find** does not display the list of files that match the search criteria.

Finding temporary files

To locate and display all files named *temp* recursively in the */usr* directory, use the following command:

```
find /usr -name temp -print | more
```

Finding files of a certain size

You can use **find** to locate files of a specified size or type. For example, to locate and print a list of all the directories (**-type d**) greater than 3 blocks in size (**-size +3**) in all the directories (**/** and below), enter:

```
find / -type d -size +3 -print
```

Finding files by permissions

Using the **-perm *onum*** option to **find**, you can locate all files with permissions that exactly match *onum* (the octal number used with **chmod(C)**). For example, to locate and display all the files in the */u* directory that give all users read, write, and execute permissions (*onum* is **0777**), enter the following command:

```
find /u -perm 0777 -print
```

Executing commands based on find output

Using the **-exec** option, you can execute a specific shell command on the files that **find** locates. The most common use of **-exec** is to locate a group of files and then remove them. For instance, when you retire a user, you can use **find** to locate all the files owned by that user, back them up, and then remove them from the system. To do this with **find**, enter:

```
find / -user edwarda -print | cpio -ovBc > /dev/rfd096  
find / -user edwarda -exec rm "{}" \;
```

The first command locates all the files owned by user *edwarda* and copies the files to an archive. The second command locates the files and then removes them. For more information on copying files to an archive, see the *cpio(C)* manual page.

You can instruct *find* to display a prompt (the command line that *find* generates) before executing the shell command on each file, by using the *-ok* option in place of *-exec*. When you use *-ok*, *find* prompts you with the generated command line:

```
<rm ... /u/edwarda/billboard >?
```

To execute the command (in this case, *rm*), enter *y*.

The command is not executed if you enter a character other than “*y*”, or if you press (Return).

Another common use of *find* with the *-exec* option is to locate all the files that belong to a particular group and change them. For example, if a user changes groups, you can use *find* to locate and change all their files to the new group:

```
find / -user edwarda -exec chgrp moms "{}" \;
```

If you retire a user and you want to transfer ownership of their files to another user, use the following command:

```
find / -user edwarda -exec chown earnestc "{}" \;
```

Using this construction to execute a command on a large group of files can be very slow because the *-exec* option forks a separate process for each file in the list. A more efficient method for doing this is to use *xargs(C)* in place of *-exec*. The *xargs* command forks fewer processes to execute the command on the entire group of files.

The following example illustrates how to use the *xargs* construction with *find*:

```
find / -user edwarda -print | xargs chown earnestc
```

This command accomplishes the same thing as the previous example, only much more efficiently.

NOTE If the syntax for the command that you want to execute with *xargs* deviates from the standard order (*command options arguments*), you must use *-exec*.

Locating core files

One useful application of *find* is to use the *-name* option to locate core and temporary files for removal.

A core file contains a copy of a terminated program. A UNIX system sometimes creates such a file when a program causes an error from which it cannot recover. A temporary file contains data created as an intermediate step during execution of a program. This file may be left behind if a program contained an error or was prematurely stopped by the user. The name of a temporary file depends on the program that created it. In most cases, the user has no use for either core or temporary files, and they can be safely removed.

When searching for core or temporary files, it is a good idea to search for files that have not been accessed for a reasonable period of time. For example, to find all core files in the */usr* directory that have not been accessed for one week (*-atime +7*), enter:

```
find /usr -name core -atime +7 -print
```

Once you locate the core files, you can remove them easily using one of the following two commands:

```
find /usr -name core -atime +7 -exec rm "{}" \;
```

```
find /usr -name core -atime +7 -print | xargs rm
```

These commands find all the core files in the */usr* directory that have not been accessed in the last seven days and remove them.

Checking and clearing log files

A UNIX system maintains a number of log files that contain information about system usage. When new information is generated, the system appends it to the appropriate log file, preserving the file's previous contents. Because some log files can rapidly become quite large, it is important to check the files periodically and, if necessary, clear them by deleting their contents. It may be necessary to retain the most recently appended information in some log files, such as */etc/lddate*. In these cases, trim the files by deleting the previous data and leaving the last entry.

Some log files only grow in special situations. For example, if you want to record all attempts to use the *su* command, and you set the *SULOG* option in */etc/default/su*, the log file */usr/adm/sulog* may build up rapidly.

Table 5-3 lists some of the log files that are most likely to need clearing or trimming. Your system may have different log files from those listed, depending on its configuration and the utilities and application software installed. You may also need to check the files more or less frequently than indicated in the table, depending on system activity. Use the *find(C)* command with the *-size* flag to locate your system's unlisted large log files.

Table 5-3 Administrative log files

Logfile path	Purpose	Checking frequency
/etc/wtmp	historical login record	weekly
/usr/adm/pacct	process accounting log file	weekly
/usr/adm/messages	system messages log file	weekly
/tcb/audittmp/*	audit system temporary files	weekly
/etc/ddate	records date of each backup	monthly or yearly
/usr/spool/uucp/LOGFILE	records of UUCP job requests, file transfers, system status	monthly
/usr/spool/uucp/.Log/.Old/*	old UUCP log files stored by uudemon.clean	monthly
/usr/spool/lp/logs/request	record of print requests	automatic†

† Maintenance of this file (and other files) can be accomplished automatically; see “Automatic log cleaning” (this page).

To clear a log file and retain file permissions:

1. Copy the file to a new filename. For example:
`cp /usr/adm/messages /usr/adm/messages.old`
2. You can clear the file by using one of the following commands:
 Bourne or Korn shell:
`> /usr/adm/messages`
 C shell:
`cat /dev/null > /usr/adm/messages`

To monitor new information that is currently being appended to system log files, use the `tail(C)` command as follows:

```
tail -f /usr/adm/messages
```

With the `-f` (follow) option, `tail` prints the last ten lines of the file, followed by any lines that are appended to the file between the time `tail` is initiated and killed with the `(Del)` key.

Automatic log cleaning

It is also possible to automate the clearing of log files. using the `cron` and `crontab` facilities. By adding a line to the `crontab(C)` file for `root`, you can excute your own file maintenance scripts on a daily, weekly, or monthly basis.

The following is one of the default entries in the *root crontabs* file (*/usr/spool/cron/crontabs/root*):

```
17 5 * * 0 /etc/cleanup > /dev/null
```

See **crontab(C)** for complete information on the format of a *crontabs* file.

The entry for */etc/cleanup* runs each Sunday morning at 5:17. The */etc/cleanup* script is reproduced below:

```
:
#
# clean up super-user log
cp /usr/adm/sulog /usr/adm/Osulog
> /usr/adm/sulog
#
# clean up volcopy log
[ -f /etc/log/filesave.log ] && mv /etc/log/filesave.log /etc/log/Ofilesave.log
> /etc/log/filesave.log
chown root /etc/log/filesave.log
chgrp sys /etc/log/filesave.log
chmod 666 /etc/log/filesave.log
#
# clean up wtmp
> /etc/wtmp
#
# clean up miscellaneous files
find / -name core -atime +7 -exec rm -f {} \;
```

/etc/cleanup does the following once per week:

- copies the *sulog* file to *Osulog* and then clears *sulog*;
- if *filesave.log* exists, moves it to *Ofilesave.log*, creates a new, empty *filesave.log* and sets the proper permissions;
- clears the */etc/wtmp* file;
- removes all *core* files that have not been accessed in the last seven days.

Maintaining efficient filesystem organization

There are two aspects of filesystem usage that degrade the efficiency of filesystems:

- Disk fragmentation (scattering of available disk space) due to constant use and reuse of filesystem blocks.
- Directories growing too large, thus increasing search time for files.

This section explains how to avoid these problems and maintain efficient filesystem organization.

Disk fragmentation

If your system has been in use for some time, the constant creation and removal of files creates a situation called *disk fragmentation*. This means that the files in the filesystem are written in small pieces on the hard disk. This increases access time and reduces system efficiency.

You can reduce disk fragmentation by first making a complete backup of all the files in the filesystem and then removing all the files from the hard disk and restoring them from the backup.

Because the files are completely rewritten on the disk during the restore, each file is written in one piece and fragmentation is reduced. A small amount of space is also recovered. It is a good idea to perform this action about once a year on a heavily used system and less often on a lightly used system. Be certain that you have complete, accurate, and readable backups before you begin or your files may be lost.

Large directories

It is wise not to allow directories to grow larger than necessary. You should be aware of several size limitations. A directory that contains entries for up to 62* files (plus the required "." and ".." entries) fits in a single disk block and can be searched very efficiently. A directory can have up to 638* entries and still be viable, as long as it is used only for data storage; anything larger is usually wholly unsuitable as a working directory. It is especially important to keep login directories small, preferably one block at most.

If large numbers of files are moved in and out of a directory, as in the case of a spool directory, it can also become oversized. This is because, as a rule, directories never shrink. Even if you delete files until you have less than 62 in your login directory, or less than 638 in a spool directory or other data collection directory, system searches of the directory slow down. This is because the directory slot remains the same size as it was prior to the removal of the files.

To see if a directory is too large, even if it does not seem to have over 62 files in it, use the command `hd | wc -l`. This tells you the number of lines (inodes) in the directory listing.

The solution to the large directory problem is twofold. Educate users to keep their login directories small, and use `cpio` to back up and restore any directory that has an excess of inode numbers left over from deleted or moved files.

* These figures apply to filenames of 14 characters or less. As filename lengths increase, up to a maximum of 255 characters, the number of files that fit on a single disk block decreases, thus reducing the optimum number of files in a directory.

Limiting login directory size

To limit the size of the login directory, each user should create a hierarchy of subdirectories that reflects the use or purpose of the contained files. This can make finding files easier for the user. Emphasize the importance of setting up a simple, clear hierarchy, with only five to 10 directories at each level, so files can be added in the future without having to change the hierarchy or move files around.

Removing excess inode numbers

To remove “shadow” inode numbers (numbers left when files are deleted or moved to subdirectories), use `cpio` to back up and restore an oversized directory. Then use `hd` to list the files in the directory and see if there are any that do not show in a standard listing. These “shadow” files appear at the end of the list, and have null inode numbers; 00 and 00 appear in the first two columns of hex digits. Use the following procedure to remove these empty file slots from the directory. Note that the same procedure can be applied to a spool directory.

1. Move to the directory above the oversized one and create a backup directory. For example, if a user named *tracy* has an oversized home directory, enter:

```
mkdir tracy.old
```

2. Move to the oversized directory and back it up:

```
find . -print | cpio -pdlm ../tracy.old
```

This command copies the directory and all subdirectories, linking files instead of copying them where possible. Confirm that the ownership, group, and permissions of the new directory and files match the original.

3. Use `l` and `hd` to check the backup directory to see that all the files are there and all the shadow inode numbers are gone.
4. Move to the directory above, then delete the oversized directory:

```
rm -rf tracy
```

5. Rename the backup directory to replace the oversized one:

```
mv tracy.old tracy
```

The directory appears the same to the user, but it can be searched more quickly. This can improve system performance, especially if the procedure is used on numerous working directories.

Adding disk space and filesystems

If free space is chronically low, it may be to your advantage to expand the system's storage capacity by installing additional hard disks as described in the *Hardware Configuration Guide*. Once the new disk is mounted, you can use the free space in the new filesystem for your work, or even copy user or system directories to it.

Suppose free space is low on your primary hard disk because one filesystem on it is full. If there are others with free space, or there is unused space on the hard disk, you can change the layout on the primary disk, but this is not as simple as adding a second hard disk. It is always best to plan the layout of your hard disk in advance as described in the *Installation and Upgrade Guide*.

A chronic shortage of space usually results from having more users on the system than the current hard disk can reasonably handle, or having too many directories or files. In either case, creating a new filesystem on a new hard disk allows some of the users and directories to be transferred from the primary hard disk, freeing a significant amount of space on the existing filesystem and improving system operation.

If you decide to change the number of filesystems on your primary hard disk or to reapportion the disk space among the filesystems, you must back up your system and reinstall it as described in the *Installation and Upgrade Guide*. You should initialize your disk manually so that you can control the layout of your disk. During the installation process, use manual layout control to reapportion your disk space as desired.

Filesystem integrity

The most important job of the operating system is to maintain the integrity of filesystem data. Actual loss of data is a rare occurrence; UNIX system filesystems are very resistant to corruption. This is because a certain amount of redundancy exists in special structures that are invisible to the user. It is these structures that ensure filesystem integrity. For example, when the system suffers a power outage, very little information is lost. Any damage usually affects one or two files, making them inaccessible. In almost all cases, the operating system can repair any damage done to files. In very rare cases, damage causes the entire filesystem to become inaccessible.

The operating system uses the `fsck` (for "filesystem check") program to repair damaged filesystems. The `fsck` program checks the consistency of filesystems. In cases where the contents of a file are lost (rare), the only way to restore lost data is from filesystem backups. `fsck` is automatically run at boot

time after an abnormal shutdown on the root filesystem. The **fsck** status messages look like this:

```
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Pathnames
** Phase 3 - Connectivity
** Phase 4 - Reference Counts
** Phase 5 - Check Free List
```

If the system terminated abnormally (for example, power outage), you see other messages that may seem alarming:

```
FREE INODE COUNT WRONG IN SUPERBLK
FIX?
```

In fact, this kind of message is routine when a system was not shut down properly, and you only have to enter **y** and **fsck** continues its recovery work. This could be done without the system administrator's intervention, but it is generally better to know what is happening to a filesystem after a problem has occurred.

To discuss the idea of filesystem integrity and how **fsck** functions, it is useful to explain the basic structure of files, directories, and filesystems. Although it is not necessary to understand the principles of file storage, it is helpful to know what the messages like the one above refer to so they will seem less mysterious.

The following section describes some of the basic principles of UNIX operating systems. Refer to "Repairing filesystems with **fsck**" (page 115) for information on how to use the **fsck** command.

How UNIX systems maintain files

Each filesystem contains special structures that allow the operating system to access and maintain the files and data stored on the filesystem. It is the disruption and repair of these structures that we are concerned with.

The structure of a filesystem is based on the way that hard disks store data. Although the hard disk contains all the data used by the system, it is not stored in neat little locations that correspond to individual files. It is unlikely that you could point to a spot on a hard disk and truthfully say: "My file is stored right there on this part of the disk." In fact, the data is probably scattered across the disk. The operating system uses a sophisticated addressing scheme to access each of the pieces that a file is broken into and to present the data to the user as a unit.

The data is spread around because the operating system does not really deal with files, but rather with units of data. For example, assume you created a file and it is actually stored on one part of the disk. Then, suppose you edit that file and delete a few sentences here and there. This means that you are

now using a little bit less disk space than when you started. This space amounts to a series of gaps in the area where your file was stored. Disk space is a precious commodity and is not wasted. So, those small amounts of disk space are allocated to other files. Picture this process on a scale of hundreds of files with a dozen users and you have an idea of how files are maintained. Because of the effectiveness of the algorithms (formulas) that the operating system uses, this process is remarkably efficient and trustworthy.

How UNIX systems maintain filesystems

A filesystem contains files and directories that are represented by special structures called “inodes” and “data blocks.” These structures make it possible for the operating system to create and keep track of filesystems.

Data blocks A block is a 1024-byte unit of data stored on the disk. A data block can contain either directory entries or file data. A directory entry consists of an inode number and a filename.

Inodes An inode can be thought of as a card from a library card catalog. Each inode contains information about a file, just as a card contains information about a book, including its location, its size, the type of file, and the number of directory entries linked to the file. One important point to remember is that an inode does not contain the name of a file; directories contain the actual names. An inode contains the locations of all the data that make up a file so the operating system can collect it all when needed.

Blocks are not just stored on the hard disk. To minimize seeking data on the hard disk, recently used data blocks are held in a cache of special memory structures called buffers. These structures make the operating system more efficient. When enough data is accumulated to write out one or more full disk blocks, the buffer is “flushed” by writing its information to the disk. A minor amount of information is always lost when an outage occurs because recently changed data has not been written to the disk.

With a hard disk filled with data, inodes, directories, files, and blocks cached in memory, how does the operating system keep track of them? The answer is that all these structures maintain sufficient connectivity between files and directories to allow severed connections to be reconstructed.

One special data block, the “super” block, contains overall information about the filesystem, rather than where a particular piece of a file is located. The super block contains the information necessary to mount a filesystem and access its data. It contains the size of the filesystem, the number of free inodes, and information about free space available.

Information is read from the disk version of the super block when the filesystem is mounted and is maintained and modified in memory as activity takes place on the system. The information is written back to the disk at regular intervals by the `init` command which is always running. The `init` command calls the `sync(ADM)` command every 60 seconds, which forces the memory version of the super block and buffers to be written to disk. If the system crashes and the information stored on the disk is not reasonably up-to-date, the filesystem might be corrupted.

Causes of filesystem corruption

Corruption can affect all the structures mentioned in this section. This means that the data or the structures used to locate data can be damaged. This can occur for several reasons:

Hardware failure	Hardware failures are rare. The best way of dealing with it is to be sure that recommended diagnostic and maintenance procedures are followed conscientiously.
Program interrupts	It is possible that errors that cause a program to fail might result in the loss of some data. It is not easy to generalize about this because the range of possibilities is so large.
Human error	While it may be painful to admit it, probably the greatest cause of filesystem corruption falls under this heading. There are rules that should be followed when managing filesystems.

Rules for checking filesystems

When checking and repairing filesystems, keep in mind the following rules:

- To make sure users cannot access the filesystem you are planning to unmount, bring the system down to single-user (maintenance) mode before unmounting it.
- Unmount the filesystem with `umount(ADM)` before checking and repairing it.
- In the case of a floppy disk filesystem, unmount it before physically removing it.
- If you plan to make the filesystem available, remember to remount the filesystem after checking and repairing it.

Regular filesystem backups represent the best assurance of continued filesystem integrity.

Repairing filesystems with *fsck*

To check and repair filesystems, use **fsck**(ADM). The **fsck** command examines the various structures on the disk and attempts to reconcile them. Where possible, **fsck** reestablishes connections and resolves references; it “cleans” the filesystem.

Before repairing a filesystem with **fsck**, shut down the system and bring it up in single-user (maintenance) mode. To do this without completely shutting down the system, use the **su** argument to **shutdown**:

```
/etc/shutdown -g10 su
```

The first argument indicates the number of minutes before system shutdown.

NOTE The **shutdown** command calls **sync** automatically.

For more information on shutting the system down, see **shutdown**(ADM).

Once the system is in single-user mode, unmount the filesystem with **umount**(ADM). Check and repair the filesystem using the following command:

```
fsck filesystem
```

Δ **sysadmsh** users select: Filesystems ⇔ Check

where *filesystem* is the name of the special file corresponding to the device name of the filesystem. For example, the */dev/u* device file corresponds to the */u* filesystem.

To check the root filesystem, use the **-b** argument to **fsck**:

```
fsck -b /dev/root
```

The system must be in single-user mode before running **fsck -b** on the root filesystem. This command automatically remounts the root filesystem after checking it.

NOTE The **fsck** program is actually a front-end that invokes a version of **fsck** for each filesystem type. For example, **fsck** calls a special version to repair DOS filesystems.

For example, if you bring your system back up in single-user mode after a power failure, use the **fsck** command to check the */u* filesystem before going into multiuser mode. To do this, enter the following command:

```
fsck /dev/u
```

The **fsck** program checks the filesystem and reports on its progress with the following messages:

```
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Pathnames
** Phase 3 - Connectivity
** Phase 4 - Reference Counts
** Phase 5 - Check Free List
```

If a damaged file is found during any one of these phases, **fsck** asks if it should be repaired or salvaged. Enter **y** to repair a damaged file. You should always allow the system to repair damaged files even if you have copies of the files elsewhere or intend to delete the damaged files.

Note that the **fsck** command deletes any file that it considers too damaged to be repaired. You can elect for **fsck** to make the repair or not. You might choose to have **fsck** ignore an inconsistency because the problem is so severe that either you want to fix it yourself using the **fsdb**(ADM) utility, or you plan to restore your system from backups. If you cannot use **fsdb**, you must allow **fsck** to resolve the inconsistencies or the filesystem may not be usable.

You may need to run **fsck** several times before the entire filesystem is clean. For a complete list of error messages, see the **fsck**(ADM) manual page.

Summary of fsck phases

The **fsck** program scans and examines each of the structures mentioned earlier. Each phase compares components and checks that these components agree with each other.

Phase 1 checks the blocks and sizes. The **fsck** program reads the inode list to determine the sizes and locates the blocks used by each file. Inodes are checked for inode type, zero link counts, inode sizes, and bad or duplicate blocks. (Bad blocks are block values outside the boundaries of a filesystem.) When **fsck** asks whether or not to clear an inode, this means to zero out the bad information in the inode. This removes the file or directory that was associated with it. A duplicate block means that two inodes point to the same block on the disk. The **fsck** command attempts to find the original inode along with the duplicate for correction in phase 2.

Phase 2 checks the pathnames. Files removed in phase 1 must then have their directory entries removed. Phase 2 cleans up error conditions caused by improper inode status, out-of-range inode pointers, and directories that point to bad inodes as described earlier. For files with duplicate blocks found in phase 1, **fsck** wants to remove both files (this is one of the few areas where system administrator intervention is useful).

Phase 3 checks for connectivity. Phase 2 removed directories that do not point to valid files. Phase 3 reconnects files that were severed from the directory structure. Any files that are unreferenced but valid are placed in a special directory called *lost+found*. Because the directory was severed, the name of the file is lost and a number is assigned to the file in *lost+found*.

NOTE `fsck` does not create or extend the *lost+found* directory. There must already be a sufficient number of empty slots in the directory for use by `fsck` when reconnecting files. A *lost+found* directory is created automatically by `mkdev fs` when a new filesystem is created.

Phase 4 checks the reference counts. The `fsck` command checks the link count of each entry that survives phases 2 and 3. In some cases, files that were not pointed to under the directory structure, but still have an inode, can be relinked to the filesystem in *lost+found*.

Phase 5 checks the free list. The `fsck` command examines the free-block list maintained by the filesystem and resolves the missing or unallocated blocks allocated or removed earlier. When an inconsistency is detected, `fsck` prompts to rebuild it.

Phase 6 salvages the free list. If specified in phase 5, the system reconstructs a free block list from the altered filesystem.

Automatic filesystem check

The operating system sometimes requests a check of the filesystem when you first start it. This usually occurs after an improper shutdown (for example, after a power loss). The filesystem check repairs any files disrupted during the shutdown.

Chapter 6

Using printers

Most systems require the ability to print out data on paper. A wide variety of printing hardware or line printers are supported. Some line printers are *parallel* devices, but most are connected as *serial* devices.

To add a printer, the system administrator must:

- connect the physical hardware to the computer, then
- use the correct system commands to enable the printer for operation.

This chapter explains how to maintain printers. Consult the *Hardware Configuration Guide* for information on connecting and configuring printers.

The operating system supports serial printers which use the standard RS-232 interface and parallel printers which use the Centronics parallel interface. To find out which interface your printer uses, check your hardware documentation.

The printer spooling system

The UNIX system line printer spooling system is a collection of commands that help you to install, monitor, and efficiently control the line printers serving your system. A request to print a file is *spooled* or lined up with other printing jobs to be sent to the printer. Each print job is processed and waits its turn in line to be printed. This line of pending print jobs is called the *queue*.

When a user requests a file to be printed using the `lp(C)` command, the line printer system responds with a "request ID." This consists of the name of the printer on which the file is printed and a unique number identifying the file. With this request ID, the user can find out the status of the print request or cancel it. The `lp` options help the user to control printer output easily.

The print service performs the following functions:

- handles the task of receiving files that users want printed
- filters the files (if necessary) so they can print properly
- schedules the work of one or more printers
- starts programs that interface with the printer(s)
- keeps track of the status of jobs
- alerts you to printer problems
- keeps track of the mounting of forms
- issues error messages when problems arise

There are several terms used in this chapter to describe the operation of the print service:

<i>device</i>	The target for lp output. A device is represented by a full pathname of a special device file.
<i>printer</i>	The name assigned by the system administrator to represent a device. This name can have up to 14 characters. At different times, a printer can be associated with different devices.
<i>class</i>	An ordered list of printers. Print requests sent to a class of printers are printed by the first available member of that class.
<i>destination</i>	A place where print requests are sent. A destination can be a class or a printer.

Consult your computer and line printer hardware manuals for information on making the connection between your system and printing devices.

Printer configuration options

Although the default values for printer configuration are usually sufficient for most needs, there are a number of options to configure individual aspects of printer operations. These include such options as fault alerting and recovery. The following is a list of additional information that can be given to define the configuration of each printer:

- interface program
- printer type
- content types
- character sets or print wheels
- fault alerting

- fault recovery
- default printing attributes

You need to give very little of this information to add a new printer to the print service; however, the more information you provide, the better the printer is managed for you and the more efficiently the print service will run.

The descriptions in the following sections help you understand what this printer configuration information means and how it is used so that you can decide how to configure your printers. In each section, you are also shown how to specify this information when adding a printer. While you can follow each of the sections in order and correctly configure a printer in several steps, you may want to wait until you have read all of the sections before adding a printer so that you can do it in one step.

Interface program

Interface programs (also known as interface scripts) are used to initialize printers and take advantage of their particular capabilities. The *standard* interface program defines minimal capabilities common to most printers. These programs are human-readable files located in */usr/spool/lp/model*. The interface programs already present on your system are listed in Table 6-1. You can select an interface program when you install or change a printer configuration using the *sysadmsh* selections

Printers ⇨ Configure ⇨ Add

or

Printers ⇨ Configure ⇨ Modify

Table 6-1 Printer interface programs

Name	Description
1640	DASI 1640 terminal
5310	AT&T 5310/20 Matrix Printer
HPDeskJet500	Hewlett Packard DeskJet 500
HPDeskJetplus	Hewlett Packard DeskJet Plus
HPLaserJet	Hewlett Packard LaserJet
TandyDMP	Tandy DMP Printers
cnlmap	serial or parallel printer requiring newline mapped to control-linefeed
dqp10	DQP-10 Matrix Printer

(Continued on next page)

Table 6-1 Printer interface programs
(Continued)

Name	Description
dumb	dumb line printer
emulator	Tandy Printers in IBM Emulation Mode
epson	Epson serial or parallel printer
f450	DASI 450 terminal
hp	hp2631a line printer
lqp40	LQP-40 Letter Quality Printer
network	remote printing over UUCP or ethernet
network.ps	remote printing over UUCP or ethernet for PostScript printers
ph.daps	Autologic APS-5 phototypesetter
postscript	PostScript printer
pprx	Printronix line printer with parallel interface
proprinter	IBM Proprinter XL
prx	Printronix line printer
qume1155	Qume Sprint 1155 line printer
standard	standard printer interface program
ti800	Texas Instruments 855 printer

Many of these interface programs have special options available by using the `-o` option to the `lp` command. (Read the appropriate interface file for this information.) For example, the PostScript interface program includes the options listed in Table 6-2.

Table 6-2 PostScript options

Option	Description
port	prints text in portrait mode
land	prints text in landscape mode
land2	prints text in 2 page landscape mode
raw	prints a PostScript file*

* Use the raw option for PostScript files only; the postscript interface script converts text files to PostScript automatically.

You can also create your own interface scripts or customize existing ones to suit your needs. Refer to “How to write an interface program” (page 156) for more information.

Printer type

The printer type is the generic name for the printer. The print service uses the printer type to extract information about the printer from the *terminfo* database. This information describes the capabilities of the printer so that you can

be warned if some of the configuration information you provide is not appropriate for the printer. The information also describes the control data to use to initialize the printer before printing a file.

NOTE Use of the *terminfo* database is not recommended for defining printer capabilities. Printer capabilities are most commonly defined through interface scripts.

The printer type is the generic name for the printer. Typically, the printer type is derived from the manufacturer's name, such as "495" for the AT&T 495 Laser Printer. Examine */usr/lib/terminfo* for an appropriate entry for your "type" of printer. There might be one entry specific to your printer, or one that is generic to a type. Specify the printer type as follows:

```
/usr/lib/lpadmin -p printername -T printer-type
```

△ *sysadmsh* users select: Printers ⇨ Configure ⇨ Parameters

If you do not define the printer type, the default **unknown** is used. This produces empty results when the print service looks up information about the printer, so the print service cannot verify certain requests or initialize the printer.

Content types

While the printer type information tells the print service what type of printer is being added, the content type information tells the print service what types of files can be printed. Most printers can print only one type of file; for them, the content type is likely to be identical to the printer type. Some printers, however, can accept several different types of file and print their contents properly. When adding this kind of printer, you should list the names of the content types it accepts.

When a file is submitted to the print service for printing, the print service searches for a printer capable of handling the job. The print service can identify an appropriate printer through either the content-type name or the printer-type name. Therefore, you can specify either name (or no name) when submitting a file for printing.

Content-type names may look a lot like printer-type names, but you are free to choose names that mean something to you and the people using the printer. (The names *simple*, *terminfo*, or *any* are recognized as having particular meanings by the print service; be sure to use them consistently.) The names must contain no more than 14 characters and may include only letters, digits, and underscores. If the same content type is printable by several different types of printer, you should use the same content type names when you add those printers. This makes it easier for the people using the printers because they can use the same name to identify the type of file they want printed regardless of the printing destination.

For example, several manufacturers produce printers that accept PostScript files. While these printers may need different printer types so that each can be properly initialized (assuming the initialization control sequences are different), they may all be capable of handling the same type of input file, which you call, perhaps, *PostScript*. As another example, several manufacturers produce printers that accept ANSI X3.64 defined escape sequences. However, the printers may not support all the ANSI capabilities or may support different sets of capabilities. You may want to give different content-type names for these printers to differentiate them.

You do not have to list the content types for a printer. If you do not, the printer type is used as the name of the content type the printer can handle. If you have not specified a printer type, the print service assumes the printer can print only files of content type *simple*. This may be sufficient if users pick the proper printer and make sure the files are properly prepared for the printer before they are submitted for printing.

The most common type of file on the UNIX system is known as *simple*. This file is assumed to contain just printable ASCII characters and the following control characters:

backspace	moves the printing mechanism back one space, except at the beginning of a line
tab	moves the printing mechanism to the next tab stop, which is normally every 8 columns on most printers
linefeed	moves the printing mechanism to the beginning of the next line (may require special port settings for some printers — refer to “Adjusting the printer port characteristics” (page 152) for more information)
form feed	moves the printing mechanism to the beginning of the next page
carriage return	moves the printing mechanism to the beginning of the same line (may fail on some printers)

The word “carriage” may be archaic for modern laser printers, but similar actions apply. If a printer can handle a *simple* type of file, you should include it in the content type list when you add the printer and specify the content type(s) that the printer can handle. If you do not want a printer to accept files of type *simple*, you must give an alternate list of content types that the printer can accept. (The printer type is a good name to use if no other type is appropriate.)

Another content type name is *terminfo*. This does not refer to a particular type of file but instead refers to all the types represented in the *terminfo* database. It is not likely that any printer is capable of handling all the types listed in the

database. However, this name is reserved for describing possible filter capabilities. Likewise, the content type *any* is reserved for describing the types of files a filter can accept or produce. These names should not be used as content types when adding a printer.

Specify the list of content types as follows:

```
/usr/lib/lpadmin -p printername -I content-type-list
```

Δ sysadmsh users select: Printers ⇌ Configure ⇌ Content

The *content-type-list* is a list of names separated by a comma or space. If you use spaces to separate the names, enclose the entire list (but not the *-I*) in quotes. If you do not define the types of files a printer can accept, the print service assumes it can take type *simple* and a type with the same name as the printer type (if the printer type is defined).

Connection method

The print service allows you to connect your printers in a variety of ways. The simplest way is to connect your printer directly to the computer. However, you may want to connect printers via a network or through a dialed modem, where they can be shared with other computers or workstations. Once you have connected the printer to the computer or connected it to a network, and connected the network to the computer, you should describe the connection method for the print service.

The default method by which printers are connected to the computer is the direct connection method. If you have used this method to connect your printer to your computer, you generally need to do only one other thing: name the connecting port.

There are two methods of making non-direct connections: through a dial-up modem or over any other type of network. The difference between a UUCP network printer and a dialup printer is that a dialup printer does not go through the UUCP spooling process; the connection is made directly and the print job prints directly on the dialed-up printer. Refer to “Configuring a network printer” (page 162) or “Configuring a dialup printer” (page 163) for more information.

Character sets or print wheels

Printers differ in the way they can print in different font styles. Some have changeable print wheels, some have font cartridges, others have preprogrammed, selectable character sets. The print service, with your help, can minimize the impact of these differences on the users of the print service.

When adding a printer, you can specify what print wheels, font cartridges, or character sets are available with the printer. Only one of these is assumed to apply to each printer. From the point of view of the print service, however, print wheels and font cartridges are the same because they require you to physically intervene and mount a new print wheel or font cartridge. Thus, for ease of discussion, only print wheels and character sets are mentioned.

When you list the print wheels or character sets available, you are assigning names to them. These names are for your convenience and the convenience of the users. Because different printers may have similar print wheels or character sets, you should use common names for all printers. This allows a person to submit a file for printing and to ask for a particular font style, without regard for which printer is used or whether a print wheel or selectable character set is used.

If the printer has mountable print wheels, you need only list their names. If the printer has selectable character sets, you need to list their names and map each one into a name or number that uniquely identifies it in the *terminfo* database. You can use the following command to determine the names of the character sets listed in the *terminfo* database:

```
TERM=printer-type tput csnm 0
```

printer-type is the name of the printer type in question. The name of the 0th character set (the character set obtained by default after the printer is initialized) should be printed. Repeat the command, using 1, 2, 3, and so on in place of the 0, to see the names of the other character sets. In general, the *terminfo* names should closely match the names used in the user documentation for the printer. However, because not all manufacturers use the same names, the *terminfo* names may differ from one printer type to the next.

NOTE For the print service to find the names in the *terminfo* database, you must specify a printer type. See the earlier section “Printer type.”

To specify a list of print wheel names when adding a printer, enter the following command:

```
/usr/lib/lpadmin -p printername -S print-wheel-list
```

Δ **sysadmsh** users select: Printers ⇨ Configure ⇨ Parameters

print-wheel-list is a list of names separated by a comma or space. If you use spaces to separate the names, enclose the entire list (but not the -S) in quotes.

To specify a list of character set names and to map them into *terminfo* names or numbers, enter the following command:

```
/usr/lib/lpadmin -p printername -S character-set-list
```

Δ **sysadmsh** users select: Printers ⇨ Configure ⇨ Parameters

character-set-list is also a list of names separated by a comma or space; however, each item in the list looks like one of the following:

```
csN=character-setname
character-setname1=character-setname2
```

N in the first case is a number from 0 to 63 that identifies the number of the character set in the *terminfo* database. *character-setname1* in the second case identifies the character set by its *terminfo* name. In either case, the name to the right of the equal sign (=) is the name you choose as an alias of the character set.

NOTE You do not have to provide a list of aliases for the character sets if the *terminfo* names are adequate. You can refer to a character set by *terminfo* name, by number, or by your alias.

For example, suppose your printer has two selectable character sets (sets #1 and #2) in addition to the standard character set (set #0). The printer type is 5310. You enter the following commands to determine the names of the selectable character sets:

```
TERM=5310 tput csnm 1
english
TERM=5310 tput csnm 2
finnish
```

The words “english” and “finnish” are the output of the commands, the names of the selectable character sets. You feel that the name “finnish” is adequate for referring to character set #2, but better names are needed for the standard set and set #1. You enter the following command to define synonyms:

```
/usr/lib/lpadmin -p printername -S "cs0=american, english=british"
```

Δ sysadmsh users select: Printers ⇨ Configure ⇨ Parameters

If you do not list the print wheels or character sets that can be used with a printer, then the print service assumes the following: a printer that takes print wheels has only a single, fixed print wheel, and people cannot ask for a special print wheel when using the printer. Also, a printer that has selectable character sets can take any *csN* name or *terminfo* name known for the printer.

Alerting to mount a print wheel

If you have printers that take changeable print wheels and you have listed the print wheels allowed on each, then users can submit a print request to use a particular print wheel. However, until it is mounted a request for a print wheel stays queued and is not printed. (Refer to “Mounting a form or print wheel” (page 139) for more information.) You could periodically monitor the number of print requests pending for a particular print wheel, but the print service provides an easier way. You can ask to be alerted when the number of requests waiting for a print wheel has exceeded some threshold.

You can choose one of several ways to receive an alert:

- You can receive an alert via electronic mail. See `mail(C)` for a description of the `mail` command.
- You can receive an alert written to whatever terminal you are logged in on. You can enable the writing of messages to your terminal with the `mesg(C)` command. (The most convenient way to do this is to add the command `mesg y` in your `.profile`, `.cshrc`, `login`, or `.kshrc` file.)
- You can receive an alert through a program of your choice.
- You can receive no alerts.

NOTE If you elect to receive no alerts, you are responsible for checking whether the proper print wheel is mounted.

In addition to the method of alerting, you can also set the number of requests that must be queued before you are alerted, and you can arrange for repeated alerts every few minutes until the print wheel is mounted. You can choose the rate of repeated alerts, or you can choose to receive only one alert per print wheel.

To arrange for alerting to the need to mount a print wheel, enter one of the following commands:

```
/usr/lib/lpadmin -S print-wheelname -A mail -Q integer -W minutes  
/usr/lib/lpadmin -S print-wheelname -A write -Q integer -W minutes  
/usr/lib/lpadmin -S print-wheelname -A 'command' -Q integer -W minutes  
/usr/lib/lpadmin -S print-wheelname -A none
```

Δ `sysadmsh` users select: Printers ⇨ Auxiliary ⇨ Alert

The first two commands direct the print service to send you a mail message or write the message directly to your terminal, respectively, for each alert. The third command directs the print service to run *command* for each alert. The shell environment currently in effect when you enter the third command is saved and restored for the execution of *command*; this includes the environment variables, user and group IDs, and current directory. The fourth command directs the print service to never send you an alert when the print wheel needs to be mounted. *integer* is the number of requests that need to be waiting for the print wheel, and *minutes* is the number of minutes between repeated alerts.

NOTE If you want mail sent or a message written to another person, you will have to use the third command listed. Use the `-A 'mail user-name'` or `-A 'write user-name'` option.

Once you start receiving repeated alerts, you can direct the print service to stop sending you alerts for the current case by giving the following command:

```
/usr/lib/lpadmin -S print-wheelname -A quiet
```

△ **sysadmsh** users select: Printers ⇌ Auxiliary ⇌ Alert

Once the print wheel is mounted and unmounted again, alerts start again if too many requests are waiting. Alerts also start again if the number of requests waiting falls below the **-Q** threshold and then rises up to the **-Q** threshold again, as when waiting requests are canceled or if the type of alerting is changed.

If *print-wheelname* is *all* in any of the commands above, the alerting condition applies to all print wheels for which an alert has already been defined.

If you do not define an alert method for a print wheel, you do not receive an alert for it. If you do define a method but do not give the **-W** option, you are alerted once for each occasion.

Fault alerting

The print service provides a framework for detecting printer faults and alerting you. Faults can range from simple problems, such as running out of paper or ribbon or needing to replace the toner, to more serious faults, such as a local power failure or printer failure. The range of fault indicators is also broad, ranging from dropping carrier (the signal that indicates that the printer is online) to sending an XOFF, or a message. Only two classes of printer fault indicators are recognized by the print service itself: a drop in carrier and an XOFF not followed in reasonable time by an XON. However, you can add filters that can recognize any other printer fault indicators and rely on the print service to alert you to a fault when the filter detects it.

NOTE Refer to “Using forms and filters” (page 148) to add a filter. For a description of how a filter should let the print service know a fault has occurred, see the **lpforms(ADM)** manual page.

You can choose one of several ways to receive an alert to a printer fault:

- You can receive an alert via electronic mail. See **mail(C)** for a description of the **mail** command.
- You can receive an alert written to the terminal on which you are logged in (any terminal). See **write(C)** for a description of the **write** command.
- You can receive an alert through a program of your choice.
- You can receive no alerts.

NOTE If you elect to receive no alerts, you need a way of finding out about the faults and fixing them; the print service does not continue to use a printer that has a fault.

In addition to the method of alerting, you can also arrange for repeated alerts every few minutes until the fault is cleared. You can choose the rate of repeated alerts, or you can choose to receive only one alert per fault.

NOTE Without a filter that provides better fault detection, the print service cannot automatically determine when a fault has been cleared except by trying to print another file. It assumes that a fault is cleared when it successfully prints a file. Until that time, if you have asked for only one alert per fault, you do not receive another alert. If after you have fixed a fault, but before the print service has tried printing another file, the printer faults again, or if your attempt to fix the fault did not succeed, you are not notified. Receiving repeated alerts per fault or requiring manual reenabling of the printer overcomes the problem (refer to “Fault recovery” (page 131) for more information).

To arrange for alerting to a printer fault, enter one of the following commands:

```
/usr/lib/lpadmin -p printername -A mail -W minutes
/usr/lib/lpadmin -p printername -A write -W minutes
/usr/lib/lpadmin -p printername -A 'command' -W minutes
/usr/lib/lpadmin -p printername -A none
```

Δ **sysadmsh usersselect:** Printers ⇌ Configure ⇌ Errors

The first two commands direct the print service to send you a mail message or write the message directly to your terminal, respectively, for each alert. The third command directs the print service to run *command* for each alert. The shell environment currently in effect when you enter the third command is saved and restored for the execution of *command*. The environment includes environment variables, user and group IDs, and current directory. The *minutes* is the number of minutes between repeated alerts. The fourth command directs the print service not to send you an alert when a fault occurs.

NOTE If you want mail sent or a message written to another person when a printer fault occurs, use the third command. Use the option:

-A '*mail username*'

or

-A '*write username*'

Once a fault occurs and you start receiving repeated alerts, you can direct the print service to stop sending you alerts for the current fault by giving the following command:

```
/usr/lib/lpadmin -p printername -A quiet
```

△ **sysadmsh users select:** Printers ⇌ Configure ⇌ Errors

If *printername* is *all* in any of the commands above, the alerting condition applies to all printers.

If you do not define an alert method, you receive mail once for each printer fault. If you do define a method but do not give the *-W* option, you are alerted once for each fault.

Fault recovery

Once a printer fault is detected and you are alerted, you will probably fix the fault and get the printer ready for printing. When the printer is ready for printing again, the print service recovers in one of three ways:

- continues printing at the top of the page where printing stopped,
- restarts printing at the beginning of the print request that was active when the fault occurred, or
- waits for you to tell the print service to re-enable the printer.

NOTE The ability to continue printing at the top of the page where printing stopped requires the use of a filter that can wait for a printer fault to be cleared before resuming properly. Such a filter probably has to have detailed knowledge of the control sequences used by the printer so it can keep track of page boundaries and know where in a file printing stopped. The default filter used by the print service cannot do this. If a proper filter is not being used, you are notified in an alert if recovery cannot proceed as you want.

To specify the way the print service should recover after a fault has been cleared, enter one of the following commands:

```
/usr/lib/lpadmin -p printername -F continue
```

```
/usr/lib/lpadmin -p printername -F beginning
```

```
/usr/lib/lpadmin -p printername -F wait
```

△ **sysadmsh users select:** Printers ⇌ Configure ⇌ Errors

These direct the print service, respectively, to continue at the top of the page, restart from the beginning, or wait for you to enter an enable command to re-enable the printer (refer to “Enabling and disabling printers” (page 137) for information on the enable command).

If you do not specify how the print service is to resume after a printer fault, it tries to continue at the top of the page where printing stopped, or failing that, at the beginning of the print request.

If the recovery is `continue` but the interface program does not stay running so that it can detect when the printer fault was cleared, printing is attempted every few minutes until it succeeds. You can force the print service to retry immediately by issuing an `enable` command.

Restricting user access to a printer

You can limit the use of a printer to a subset of all people on your computer. You may want to do this, for instance, if a printer is being set aside for printing sensitive information and only a subset of the people are allowed to print sensitive information, or if use of a high-quality printer incurs expenses not all people are authorized to incur.

The print service uses the list of users allowed or denied for a printer to restrict use of the printer. The print service refuses a user's request to print a file on a printer he or she is not allowed to use.

The method of listing the users allowed or denied for a printer is similar to the method used to list users allowed or denied access to the `cron` and `at` facilities. Briefly, the rules are as follows:

- An allow list contains those users allowed to use the printer. A deny list contains those users denied access to the printer.
- If the allow list is not empty, the deny list is ignored. If the allow list is empty, the deny list is used. If both lists are empty, there are no restrictions on who can use the printer.
- Putting "any" or "all" into the allow list allows everybody to use the printer; putting "any" or "all" into the deny list denies everybody, except the user `lp` and the super user (`root`).

You can add names of users to either list using one of the following commands:

```
/usr/lib/lpadmin -p printername -u allow:user-list  
/usr/lib/lpadmin -p printername -u deny:user-list
```

Δ `sysadmsh` users select: Printers ⇌ Configure ⇌ Users

user-list is a list of names of users separated by a comma or space. If you use spaces to separate the names, enclose the entire list (including `allow:` or `deny:` but not the `-u`) in quotes. The first command adds the names to the allow list and removes them from the deny list. The second command adds the names to the deny list and removes them from the allow list. Using `allow:all` allows everybody; using `deny:all` denies everybody.

If you do not add user names to the allow or deny lists, the print service assumes that everybody can use the printer.

Default printing attributes

When a user submits a request to print a file, the page size, character pitch, and line pitch (that is, print spacing) are normally determined from the form that it is printed on. If the user does not require a form, he or she can give the page size and print spacing to use. However, if the user gives neither a form to use nor the page size and print spacing, defaults are used.

You can set the defaults for each printer. This can also serve to make submitting a print request easier, by designating different printers as having different default page sizes or print spacing. Users then simply route their file to the appropriate printer to get the style output they want. For example, you can have one printer dedicated to printing wide (132 column) output, another printing normal (80 column by 66 lines) output, yet another printing letter quality (12 characters per inch, 8 lines per inch).

You can independently specify four default settings: page width, page length, character pitch, and line pitch. You can scale these to fit your needs. The first two can be given in columns and lines, inches, or centimeters. The last two can be given as characters and lines per inch or per centimeter. In addition, the character pitch can be specified as **pica** for 10 characters per inch (cpi), **elite** for 12 cpi, or **compressed** for the maximum cpi the printer can provide (up to a limit of 30 cpi).

Set the defaults using one or more of the following commands:

```
/usr/lib/lpadmin -p printername -o width=scaled-number
/usr/lib/lpadmin -p printername -o length=scaled-number
/usr/lib/lpadmin -p printername -o cpi=scaled-number
/usr/lib/lpadmin -p printername -o lpi=scaled-number
```

Δ **sysadmsh** users select: Printers ⇨ Configure ⇨ Parameters

Add the letter “i” to *scaled-number* to indicate inches, or the letter “c” to indicate centimeters. The letter “i” for character pitch (cpi) or line pitch (lpi) is redundant. You can also give **pica**, **elite**, or **compressed** instead of a number for the character pitch.

If you do not provide defaults, the page size and print spacing are those available when the printer is initialized. You can find out what the defaults are by first defining the printer configuration without providing your own defaults, then using the **lpstat** program to display the printer configuration. The command

```
lpstat -p printername -l
```

reports the default page size and print spacing. If you have not provided the defaults, the reported defaults are calculated from the *terminfo* database entry for the printer. Obviously, this requires you to have provided a printer type in the printer configuration.

Print service command summary

The print service has four user-level commands, which are shown in Table 6-3.

Table 6-3 User print service commands

Command	Description
cancel(C)	cancels a request for a file to be printed
lp(C)	sends a file or files to a printer
lpstat(C)	reports the status of the print service
lprint(C)	prints from printer attached to a terminal

In addition to sending requests to the print service system, checking the status of requests, and canceling requests, users can be given the ability to disable and enable a printer. The idea is that if a user finds a printer is malfunctioning in some way, it should not be necessary to call the administrator to turn the printer off. On the other hand, it may not be reasonable in your printing environment to allow regular users to disable a printer. You can control whether other users have access to the two commands shown in Table 6-4 by assigning or revoking the *printerstat* authorization. Refer to “Changing user authorizations” (page 74) for more information.

Table 6-4 Privileged print service commands

Command	Description
disable(C)	deactivates the named printer(s)
enable(C)	activates the named printer(s)

A separate set of commands available for the administrator is shown in Table 6-5. These commands are found in the */usr/lib* directory. If you expect to use them frequently, you might find it convenient to include that directory in your *PATH* variable. To use the administrative commands, you must be logged in as either *root* or have the *lp* authorization. Refer to “Changing user authorizations” (page 74) for more information. (*lpsched* must be run by *root*.)

Note that all these commands are accessible under the `sysadmsh` Printers selection. You will also probably need to use the commands for disabling and enabling a printer and the rest of the commands described earlier in this section.

Table 6-5 Administrative print service commands

Command	Description
<code>/usr/lib/accept</code>	Permits job requests to be queued for a specified destination.
<code>/usr/lib/reject</code>	Prevents jobs from being queued for a specified destination. Described on the <code>accept(ADM)</code> manual page.
<code>/usr/lib/lpadmin</code>	Sets up or changes printer configurations.
<code>/usr/lib/lpfilter</code>	Sets up or changes filter definitions.
<code>/usr/lib/lpforms</code>	Sets up or changes preprinted forms (Enter <code>/usr/lib/lpadmin</code> to mount a form).
<code>/usr/lib/lpmove</code>	Moves output requests from one destination to another. Described on the same manual page as <code>lpsched(ADM)</code> .
<code>/usr/lib/lpsched</code>	Starts the print service.
<code>/usr/lib/lpshut</code>	Stops the print service. Described on the <code>lpsched(ADM)</code> manual page.
<code>/usr/lib/lpusers</code>	Sets or changes the default priority and priority limits the users of the print service can request.

These commands are also accessed through the `sysadmsh` Printers selection, which is much easier than the complex syntax of the print service commands.

Starting and stopping the print service

Under normal operation, you should never have to start or stop the print service manually. It automatically starts each time the system goes into multi-user mode. However, if you need to stop the print service without stopping the operating system as well, you can do so by following the procedure described in the next section.

Stopping the print service causes all printing to cease within seconds. Any print requests that have not finished printing are printed in their entirety after the print service restarts. The printer configurations, forms, and filters in effect when the print service is stopped are restored after it is restarted.

NOTE To start and stop the print service manually, you must be logged in as *root*.

Manually stopping the print service

To stop the print service manually, enter the following command:

```
/usr/lib/lpshut
```

△ **sysadmsh users select:** Printers ⇄ Schedule ⇄ Stop

This message is displayed:

```
Print services stopped
```

All printing ceases within a few seconds. If you try to stop the print service when it is not running, you see the message:

```
Print services already stopped
```

NOTE Jobs can appear to pass through a printer that is not online. If a printer is not online or operating properly, you should disable the printer.

Manually starting the print service

To restart the print service manually, enter the following command:

```
/usr/lib/lpsched
```

△ **sysadmsh users select:** Printers ⇄ Schedule ⇄ Begin

This message is displayed:

```
Print services started
```

It may take a minute or two for the printer configurations, forms, and filters to be reestablished before any saved print requests start printing. If you try to restart the print service when it is already running, you see the message:

```
Print services already active
```

NOTE You do not have to stop the print service to change printer configurations or to add forms or filters.

Canceling a print request

To cancel a printout you have requested, use the `cancel(C)` command. When you request a printout, the system displays a request ID for your job. For example, if you send a job to a printer named "laser," the UNIX system displays the request ID as:

```
request id is laser-number
```

where *number* is the number assigned to your job. To cancel the job before it begins printing, use the following command:

```
cancel laser-number
```

Δ `sysadmsh` users select: Printers ⇌ Request ⇌ Cancel

The printout is canceled.

Most systems print quickly, so a `cancel` command must be used promptly to have any effect.

Enabling and disabling printers

The `enable` command allows `lpsched` to print files on printers. A printer can accept requests for printing after the `accept` command is given for it, but to print the files, the `enable` command must be given as well.

For example, to enable a printer named "daisy," enter:

```
enable daisy
```

Δ `sysadmsh` users select: Printers ⇌ Schedule ⇌ Enable

You can disable printers with the `disable` command. The scheduler, `lpsched`, does not send printing requests to disabled printers regardless of their status with respect to the `accept` command. The `disable -r` option allows you to include a message about why the printer was disabled in the output for `lpstat -t`.

For example, to disable a printer named "laser" because of a paper jam, enter:

```
disable -r "paper jam" laser
```

Users requesting the status of "laser" with the command `lpstat -plaser` receive the following message:

```
printer laser disabled since Dec 5 10:15  
paper jam
```

For more information on these two commands, see the `enable(C)` and `disable(C)` manual pages.

Adding a printer to a class

It is occasionally convenient to treat a collection of printers as a single class. The benefit is that a person can submit a file for printing by a member of a class, and the print service picks the first printer in the class that it finds free. This allows faster turnaround, as printers are kept as busy as possible.

Classes are not needed if the only purpose is to allow a user to submit a print request by type of printer. The `lp -T type` command lets a user submit a file and specify its type. The first available printer that can handle the type of file prints the file.

The print service avoids using a filter, if possible, by choosing a printer that can print the file directly over one that needs it filtered first.

One use of classes is to put into a class a series of printers that should be used in a particular order. If you have a high-speed printer and a low-speed printer, for instance, you probably want the high-speed printer to handle as many print requests as possible, with the low-speed printer reserved for use when the other is busy. Because the print service always checks for an available printer in the order that the printers were added to a class, you could add the high-speed printer to the class before the low-speed printer and let the print service route print requests in the order you wanted.

Add a printer to a class using the following command:

```
/usr/lib/lpadmin -p printername -c classname
```

△ `sysadmsh` users select: Printers ⇨ Configure ⇨ Modify

If the class *classname* does not exist yet, it is created. (Use `lpstat -c` to get a list of all classes and their printers.)

NOTE Class names and printer names must be unique. This allows a user to specify the destination for a print request without having to know whether it is a class of printers or a single printer. Thus, you can not have a class and printer with the same name.

A printer does not belong to any class until you add it to one.

Setting the system default destination

You can define the printer or class used to print a file when the user has not explicitly asked for a particular destination and has not set the `LPDEST` shell variable. The printer or class must already exist first.

Make a printer or class the default destination by entering the following command:

```
/usr/lib/lpadmin -d printername or classname
```

Δ sysadmsh users select: Printers ⇨ Configure ⇨ Default

If you later decide that there should be no default destination, enter a null *printername* or *classname* as in the following command:

```
/usr/lib/lpadmin -d
```

Δ sysadmsh users select: Printers ⇨ Configure ⇨ Default

If you do not set a default destination, there is none. Users must explicitly name a printer or class in each print request, or they have to set the `LPDEST` shell variable with the name of a destination.

For C-shell:

```
setenv LPDEST printer
```

For Bourne or Korn shells:

```
LPDEST=printer ; export LPDEST
```

Users can also place these commands in their `.login` and `.profile` files, respectively.

Mounting a form or print wheel

NOTE Refer to “Using forms and filters” (page 148) for information about preprinted forms.

Before the print service starts to print files that need a preprinted form or print wheel, you have to mount it on a printer. (Print wheels are used on older, so-called “daisy wheel” impact printers that use small wheels with the print

characters around the perimeter.) If alerting has been set on the form or print wheel, you are alerted when enough print requests are queued for it to be mounted.

When you mount a form, you may wish to see if it is lined up properly. If an alignment pattern is registered with the form, you can ask that this be repeatedly printed until you have adjusted the printer so that the alignment pattern looks correct.

Mounting a form or print wheel involves first loading it onto the printer and then telling the print service that it is mounted. Because it is difficult to do this on a printer that is currently printing and because the print service continues to print files not needing the form on the printer, you will probably have to disable the printer first. Thus, the proper procedure is as follows:

1. Disable the printer using the **disable** command.
2. Mount the new form or print wheel as described later in this section.
3. Re-enable the printer using the **enable** command. Refer to "Enabling and disabling printers" (page 137) for more information on the **enable** and **disable** commands.

After loading the new form or print wheel into the printer, enter the following command to tell the print service to mount it. (This command is shown on two lines for readability; it must be entered as one line.)

```
/usr/lib/lpadmin -p printername -M -S print-wheelname  
-f formname -a -o filebreak
```

△ **sysadmsh** users select: Printers ⇨ Auxiliary ⇨ PPforms ⇨ Configure

Leave out **-S *print-wheelname*** if you are mounting just a form, or leave out the **-f *formname* -a -o filebreak** if you are mounting just a print wheel.

If you are mounting a form, you are asked to press the (Return) key before each copy of the alignment pattern is printed. After the pattern is printed, you can adjust the printer and press the return key again. If no alignment pattern is registered, you are not asked to press the key. You can drop the **-a** and **-o filebreak** options if you do not want to bother with the alignment pattern.

The **-o filebreak** option tells the print service to add a *formfeed* after each copy of the alignment pattern. The actual control sequence used for the *formfeed* depends on the printer involved and is obtained from the *terminfo* database. If the alignment pattern already includes a formfeed, leave out the **-o filebreak** option.

If you want to unmount a form or print wheel, use the following command:

```
/usr/lib/lpadmin -p printername -M -S none -f none
```

Δ sysadmsh users select: Printers ⇨ Auxiliary ⇨ PPforms ⇨ Remove

Leave out **-S none** if you just want to unmount a form; leave out **-f none** if you just want to unmount a print wheel.

Until you mount a form on a printer, only print requests that do not require a form are sent to it. Likewise, until you mount a print wheel on a printer, only print requests that do not require a particular print wheel are sent to it.

Removing a printer or class

You can remove a printer or class if it has no pending print requests. If there are pending requests, you have to first move them to another printer or class using the **lpmove** command, or remove them using the **cancel** command.

Removing the last remaining printer of a class automatically removes the class as well. However, the removal of a class does not cause the removal of printers that were members of the class. If the printer or class removed is also the system default destination, the system no longer has a default destination.

To remove a printer or class, enter the following command:

```
/usr/lib/lpadmin -x printername or classname
```

Δ sysadmsh users select: Printers ⇨ Configure ⇨ Remove

If all you want to do is remove a printer from a class but not delete the printer, enter the following command:

```
/usr/lib/lpadmin -p printername -r classname
```

Δ sysadmsh users select: Printers ⇨ Configure ⇨ Modify

Managing the printing load

Occasionally, you may need to stop accepting print requests for a printer or move print requests from one printer to another. There are various reasons for doing this, such as the following:

- The printer needs periodic maintenance.
- The printer is broken.
- The printer was removed.
- The configuration was changed so that the printer can be used differently.

- Too many large print requests are queued for one printer and should be evenly distributed.

If you are going to make a big change in the way a printer is used, such as stopping its ability to handle a certain form, changing the print wheels available for it, or disallowing some people from using it, print requests that are currently queued for printing on it must be moved or canceled. The print service attempts to find alternate printers, but only if the user does not care which printer is used. Such requests are not automatically moved; if you do not move them first, the print service cancels them.

If you decide that a printer is to be taken out of service, its configuration is to be changed, or it is too heavily loaded, you can move print requests from it and reject additional requests for it. Use the `lpmove` and `reject` commands for this. If you do reject requests for a printer, you can later accept requests using the `accept` command.

Rejecting requests for a printer or class

To stop accepting any new requests for a printer or a class of printers, enter the following command:

```
/usr/lib/reject -r "reason" printername or classname
```

Δ `sysadmsh` users select: Printers ⇨ Schedule ⇨ Reject

You can reject requests for several printers or classes in one command by listing their names on the same line, separating the names with spaces. The *reason* is displayed whenever anyone tries to print a file on the printer. You can omit this option if you do not want to give a reason.

Although the `reject` command stops any new print requests from being accepted, it does not move or cancel any requests currently queued for the printer. These continue to print as long as the printer is enabled.

Accepting requests for a printer or class

The `accept` command allows printers or classes of printers to accept print requests made with the `lp` command. You can allow a printer to accept requests after it has been properly configured.

After the condition that led to denying requests is corrected or changed, enter the following command to start accepting new requests:

```
/usr/lib/accept printername or classname
```

Δ `sysadmsh` users select: Printers ⇨ Schedule ⇨ Accept

Again, you can accept requests for several printers or classes in one command by listing their names on the same line. You will always have to use the `accept` command for a new printer or class after you have added it because the print service does not initially accept requests for new printers or classes.

Moving requests to another printer

If you have to move requests from one printer or class to another, enter one of the following commands:

```
/usr/lib/lpmove request-id printername
/usr/lib/lpmove printername1 printername2
```

△ `sysadmsh` users select: Printers ⇌ Request ⇌ Move

You can give more than one request ID before the printer name in the first command. The first command moves the listed requests to the named printer. The latter command moves all requests currently queued for the first printer to the second printer. When the latter command is used, the print service also no longer accepts requests for the first printer (this has the same effect as the `reject` command).

Managing queue priorities

The print service provides a simple priority mechanism that people can use to adjust the position of a print request in the queue. Each print request can be given a priority level by the person who submits it; this is a number from 0 to 39, with *lower* numbers indicating *higher* levels of priority. Requests with higher priority (smaller numbers) are placed ahead of requests with lower priority (larger numbers).

In this way, if you decide that your print request is of too low a priority, you can set a higher priority (lower value) when you submit the file for printing. If you decide that your print request is of too high a priority, you can set a lower priority (higher value) when you submit the file for printing.

A priority scheme this simple does not work if there are no controls on how high one can set the priority. You can define the following characteristics of this scheme:

- Each user can be assigned a priority limit. One cannot submit a print request with a priority higher than his or her limit, although one can submit a request with a lower priority.
- A default priority limit can be assigned for the balance of users not assigned a personal limit.

- A default priority can be set. This is the priority given to print requests to which the user does not assign a priority.

By setting the characteristics according to your needs, you can prevent lower priority printing tasks (such as regular printing by most staff members) from interfering with higher priority printing tasks (such as payroll check printing by the accounting staff).

You may find that you want a critical print request to print ahead of any others, perhaps even if it has to preempt the currently printing request. You can have the print service give **immediate** handling to a print request and put on hold another print request. This lets the urgent print request print and delays the current print request until you have it **resumed**.

The `lpusers` command lets you assign both priority limits for users and priority defaults. In addition, you can use the `lp -i request-id -H hold` and `lp -i request-id -H immediate` commands to put a request on hold or move it up for immediate printing, respectively. These commands are discussed in detail in the sections that follow.

Setting priority limits

To set a user's priority limit, enter the following command:

```
/usr/lib/lpusers -q priority-level -u username
```

You can set the limit for a group of users by listing their names after the `-u` option. Separate multiple names with a comma or space (enclose the list in quotes if you use a space). The *priority-level* is a number from 0 to 39. As mentioned before, the lower the number, the higher the priority, or, in this case, the priority limit.

If you want to set the priority limit for all other users, enter the following command:

```
/usr/lib/lpusers -d priority-level
```

Δ `sysadmsh` users select: Printers ⇄ Priorities ⇄ Default

This sets the default limit; the default applies to those people who have not been given a personal limit using the earlier `lpusers` command.

If you later decide that someone should have a different priority limit, just re-enter the first command above with a new limit. If you decide that someone with a personal limit should have whatever the default limit is, enter the following command:

```
/usr/lib/lpusers -u username
```

Δ sysadmsh users select: Printers ⇌ Priorities ⇌ Remove

Again, you can do this for more than one person at a time by giving a list of names. Using the `lpusers` command with just the `-u` option puts the users in the *default limit* category.

If you do not set a default limit, people without personal limits are limited to priorities in the range of 20 to 39.

Setting a default priority

You can set the default priority that should be assigned to those print requests submitted without a priority. Use the following command:

```
/usr/lib/lpusers -q priority-level
```

Δ sysadmsh users select: Printers ⇌ Priorities ⇌ Highest

Do not confuse this default with the *default limit*. This default is applied when a user does not give a priority; the *default limit* is applied if you have not assigned a limit for a user — it is used to limit the user from giving too high a priority.

NOTE If the default priority is greater than the limit for a user, the limit is used instead.

If you do not set a default priority, the print service uses the default of 20.

Examining the priority limits and defaults

You can examine all the settings you have assigned for priority limits and defaults by entering the following command:

```
/usr/lib/lpusers -l
```

Δ sysadmsh users select: Printers ⇌ Priorities ⇌ List

Moving a request around in the queue

Once a user has submitted a print request, you can move it around in the queue to some degree. For example, you can:

- adjust the priority to any level regardless of the limit for the user

- put it on hold and let other requests print ahead of it
- put it at the head of the queue for immediate printing

You use the regular `lp` user command to do each of these.

Changing the priority for a request

Print requests that are still waiting to print can be reassigned a new priority. This repositions the request in the queue, putting it ahead of lower priority requests but behind any others at the same or higher priority. The priority limit assigned to the user (or the default priority limit) has no effect because you override this limit as the administrator.

Enter the following command to change the priority of a request:

```
lp -i request-id -q new-priority-level
```

You can change only one request at a time with this command. If a request is already printing, you cannot change its priority.

Putting a request on hold

Any request that has not finished printing can be put on hold. You can stop its printing, if it currently is printing, and keep it from printing until you resume it. Another user, however, cannot resume a print request that you put on hold.

Enter the following command to place a request on hold:

```
lp -i request-id -H hold
```

Enter the following command to resume the request:

```
lp -i request-id -H resume
```

Once resumed, a request continues to move up the queue and will print. If the request was printing when you held it, it is restarted and becomes the next request to print. Normally the request starts printing from the beginning, with page one, but you can have it start printing at a later page. Enter the following command to resume the request at a different page:

```
lp -i request-id -H resume -P starting-page-
```

The final dash is needed to specify the starting page and all subsequent pages.

NOTE The ability to print a subset of pages requires the presence of a filter that can handle this. The default filter used by the print service cannot handle it. An attempt to resume a request on a later page is rejected if an appropriate filter is not being used.

Moving a request to the head of the queue

You can move a print request to the head of the queue, where it is the next job eligible for printing. If it must start printing immediately, but another request is currently printing, you can hold the other request as described previously.

Enter the following command to move a print request to the head of the queue:

```
lp -i request-id -H immediate
```

Only the system administrator can move a request like this; regular users cannot use the **-H immediate** option.

NOTE If you set more than one request for immediate printing, they print in the reverse order set; that is, the request moved to the head of the queue most recently prints first.

Examining a printer configuration

Once you define a printer configuration, you probably want to review it to see if it is correct. If after examining the configuration you find you made a mistake, you can run the necessary command (or `sysadmsh` selection) for the characteristic concerned; you do not have to redefine the entire configuration.

Use the `lpstat` command to examine both the configuration and the current status of a printer. A short form of this command gives just the status; you can use it to see if the printer exists and if it is busy, idle, or disabled. A long form of the command adds the complete configuration. Enter one of the following commands to examine a printer:

```
lpstat -p printername  
lpstat -p printername -l
```

The second command is the long form. With either command you should see something like the following:

```
printer printer-name now printing request-id. enabled  
since date.
```

```
printer printer-name is idle. enabled since date.
```

```
printer printer-name disabled since date.  
reason
```

```
printer printer-name waiting for auto-retry.  
reason
```

The “waiting for auto-retry” output shows that the print service failed in trying to use the printer (because of the *reason* shown) and that the print service will try again later.

With the long form of the command, you may also see the following items on the output:

```
Form mounted: form-name
Content types: content-type-list
Printer type: printer-type
Description: comment
Connection: connection-info
Interface: path-name
On fault: alert-method
After fault: fault-recovery
Users allowed:
    user-list
Forms allowed:
    form-list
Banner required
Character sets:
    character-set-list
Default pitch: integer CPI, integer LPI
Default page size: scaled-decimal-number wide,
    scaled-decimal-number long
Default port settings: stty-option-list
```

Using forms and filters

A preprinted form is a paper image of a blank form that you can load into your printer. An application typically generates a file that, when printed on the blank form, fills out the form. The print service includes facilities to create and administer forms.

The print service also permits the use of filters, which are used to accomplish three things:

- Convert a user's file into a data stream that prints properly on a given printer.
- Handle the various modes of printing that people may request with the `-y` option to the `lp` command, such as two-sided printing, landscape printing, draft or letter quality printing, and so on.
- Detect printer faults and inform the print service so that the latter can alert you.

There are very few cases where it will be necessary to use forms and filters, but these features are supported. For complete information, consult the `lpfilter(ADM)` and `lpforms(ADM)` manual pages.

Using the information in the request log

The directories `/usr/spool/lp/temp` and `usr/spool/lp/requests` contain files that describe each request that has been submitted to the print service. Each request has two files, one in each directory, that contain information about the request. The information is split to put more sensitive information in the `/usr/spool/lp/requests` directory where it can be kept secure. The request file in `/usr/spool/lp/temp` is safe from all except the user who submitted the request, while the file in `/usr/spool/lp/requests` is safe from even the submitting user.

These files remain in their directories only as long as the request is on the queue. Once the request is finished, the information in the files is combined and appended to the file `/usr/spool/lp/logs/requests`.

The request log has a simple structure that makes it easy to extract data using common UNIX system shell commands. The requests are listed in the order in which they were printed and are separated by lines that give the request ID. Each line below the separator line is marked with a single letter that identifies the kind of information contained in the line. Each letter is separated from the data by a single space. Table 6-6 lists the log file codes.

Table 6-6 Request log entries

Letter	Content of line
=	This is the separator line, containing the request ID, the user and group IDs of the user, the total number of bytes in the original (unfiltered) files, and the time when the request was queued. These items are separated by commas and are in the order just named. The user ID, group ID, and sizes are preceded by the word uid , gid , and size , respectively.
C	The number of copies printed.
D	The printer or class destination or the word any .
F	The name of the file printed. This line is repeated for each file printed, and files are printed in the order given.
f	The name of the form used.
H	The type of special handling used, spelled out (resume , hold , immediate). The only useful value found in this line is immediate .
N	The type of alert used when the print request is successfully completed. The type is the letter M if the user was notified by mail, or W if the user was notified by a message to his or her terminal.

(Continued on next page)

Table 6-6 Request log entries
(Continued)

Letter	Content of line
O	The -o options.
P	The priority of the print request.
p	The list of pages printed.
r	This single letter line is present if the user asked for raw processing of the files (the -r option of the lp command.)
S	The character set or print wheel used.
s	The outcome of the request as a combination of individual bits expressed in hexadecimal form. While several bits are used internally by the Spooler, the most important bits are listed below: 0x0004 Slow filtering finished successfully. 0x0010 Printing finished successfully. 0x0040 The request was canceled. 0x0100 The request failed filtering or printing.
T	The title placed on the banner page.
t	The type of content found in the file(s).
U	The name of the user who submitted the print request.
x	The slow filter used for the request.
Y	The list of special modes to give to the filters used to print the request.
y	The fast filter used for the request.
z	The printer used for the request. This differs from the destination (the D line) if the request was queued for any printer or a class of printers or if the request was moved to another destination by the print service administrator.

Customizing the print service

Although the print service tries to be flexible enough to handle most printers and printing needs, it cannot be complete. You may buy a printer that does not quite fit into the way the print service handles printers or may have a printing need that the standard features of the print service do not accommodate.

You can customize the print service in a few ways. This section tells you how you can:

- adjust the printer port characteristics
- adjust the *terminfo* database
- write an interface program

The diagram in Figure 6-1 gives an overview of the processing of a print request.

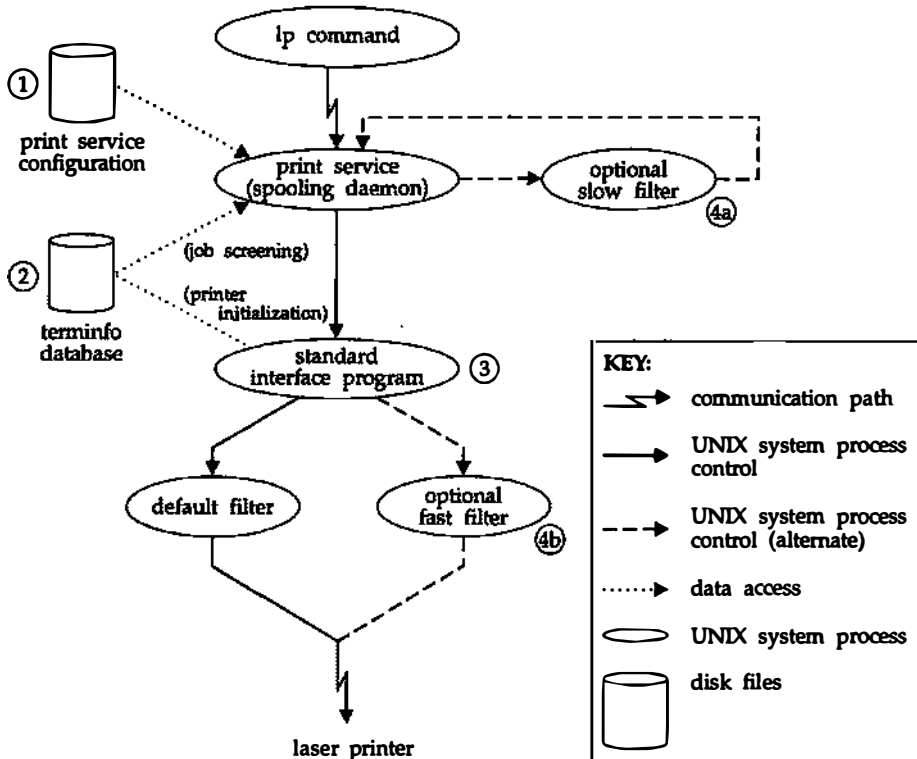


Figure 6-1 How the print service processes print request `lp -d laser file`

Each print request is sent to a *spooling daemon* (background program) that keeps track of all the requests. The daemon is created when you start the print service. This UNIX system process is also responsible for keeping track of the status of the printers and slow filters; when a printer finishes printing a user's file, the daemon starts it printing another request, if one is queued.

You can customize the print service by adjusting or replacing some of the pieces shown in Figure 6-1. (The numbers are keyed to the diagram.)

1. For most printers, you need only change the printer configuration stored on disk. The earlier sections of this chapter have explained how to do this. Some of the more printer-dependent configuration data are the printer port characteristics: baud rate, parity, and so on.
2. For printers that are not represented in the *terminfo* database, you can add a new entry that describes the capabilities of the printer. This database is used in two parallel capacities: screening print requests to ensure that those accepted can be handled by the desired printer and setting the printer so it is ready to print the request.

For instance, if the *terminfo* database does not show a printer capable of setting a page length requested by a user, the spooling daemon rejects the request. On the other hand, if it does show it capable, then the same information is used by the interface program to initialize the printer.

3. For particularly difficult printers, or if you want to add features not provided by the delivered print service, you can change the standard interface program. This program is responsible for managing the printer: it prints the banner page, initializes the printer, and invokes a filter to send copies of the user's files to the printer.
- 4a,b. To provide a link between the applications used on your system and the printers, you can add slow and fast filters. Each type of filter can convert a file into another form, mapping one set of escape sequences into another, for instance, and can provide special setup by interpreting print modes requested by a user. Slow filters are run separately by the daemon to avoid tying up a printer. Fast filters are run so their output goes directly to the printer; thus, they can exert control over the printer.

Adjusting the printer port characteristics

You should make sure that the printer port characteristics set by the print service match the printer communication settings. The standard printer port settings were designed to work with typical UNIX files and many printers, but they do not work with all files and printers. This is not really a customizing step, because a standard feature of the print service is to allow you to specify the port settings for each printer. However, it is an important step in getting your printer to work with the print service, so it is described in more detail here.

When you add a new printer, read the documentation that comes with it so that you understand what it expects from the host (the print service). Then read the manual page for the `stty(C)` command. It summarizes the various characteristics that can be set on a terminal or printer port.

Only some of the characteristics listed in the `stty(C)` manual page are important for printers. The ones likely to be of interest to you are listed in the following table (but you should still consult the `stty(C)` manual page for others).

Printers connected directly to computers and those connected over some networks require that the printer port characteristics be set by the interface program. These characteristics define the low-level communications with the printer. Included are the baud rate; use of XON/XOFF flow control; 7, 8, or other bits per byte; style of parity; and output post-processing. The standard interface program uses the `stty` command to initialize the printer port, minimally setting the baud rate and a few other default characteristics.

The default characteristics applied by the standard interface program are listed in Table 6-7

Table 6-7 Default `stty` options

Default	Meaning
9600	9600 baud rate
cs8	8-bit bytes
-cstopb	1 stop bit per byte
-parenb	no parity generation
ixon	enable XON/XOFF flow control
-ixany	allow only XON to restart output
opost	post-process data stream as listed below
-oluc	do not map lowercase to uppercase
onlcr	map linefeed into carriage-return/linefeed
-ocrnl	do not map carriage-return into linefeed
-nocr	output carriage-returns even at column 0
nl0	no delay after linefeeds
cr0	no delay after carriage-returns
tab0	no delay after tabs
bs0	no delay after backspaces
vt0	no delay after vertical tabs
ff0	no delay after formfeeds

You may find that the default characteristics are sufficient for your printers. However, printers vary enough that you are likely to find that you have to set different characteristics. See the `stty(C)` manual page to find the complete list of characteristics.

If you have a printer that requires printer port characteristics other than those handled by the `stty` program, you must customize the interface program.

When you add a new printer, you can specify an additional list of port characteristics that should be applied when printing each user's file. The list you give will be applied after the default list so that you do not need to include default items that you do not want to change in your list. Specify the additional list as follows:

```
/usr/lib/lpadmin -p printer-name -o "stty=stty-option-list"
```

Δ **sysadmsh** users select: Printers ⇌ Configure ⇌ Parameters

Note that both the double quotes and single quotes are needed if you give more than one item in the *stty-option-list*. If you do not include alternate printer port characteristics, the default list in the table will be used.

As one example, suppose your printer is to be used for printing graphical data, where linefeed characters should be output alone without an added carriage-return. You would enter the following command:

```
/usr/lib/lpadmin -p printer-name -o "stty=onlcr"
```

Note that the single quotes are omitted because there is only one item in the list.

As another example, suppose your printer requires odd parity for data sent to it. You would enter the following command:

```
/usr/lib/lpadmin -p printer-name -o "stty=parenb parodd cs7"
```

Adjusting the terminfo database

The print service relies on a standard interface and the *terminfo* database to initialize each printer and set up a selected page size, character pitch, line pitch, and character set. Thus, it is usually sufficient to have the correct entry in the *terminfo* database (*/usr/lib/terminfo/terminfo.lp*) to add a new printer to the print service. Several entries for popular printers are delivered in *terminfo* database entries with the print service package.

Each printer is identified in the *terminfo* database with a short name; this kind of name is identical to the kind of name used to set the **TERM** shell variable. For instance, the AT&T model 455 printer is identified by the name "455".

If you cannot find a *terminfo* entry for your printer, you should add one. If you do not, you may still be able to use the printer with the print service, but you cannot get automatic selection of page size, pitch, and character sets, and you may have trouble keeping the printer set in the correct modes for each print request. Another option to follow instead of updating the *terminfo* entry is to customize the interface program used with the printer. See the next section for details on how to do this.

There are hundreds of items that can be defined for each terminal or printer in the *terminfo* database. However, the print service uses less than fifty of these, and most printers need even less than that. Table 6-8 lists the items that need to be defined (as appropriate for the printer) to add a new printer to the print service.

Table 6-8 terminfo entry definitions for printers

terminfo item	Meaning
Booleans:	
daisy	printer needs operator to change character set
Numbers:	
bufsz	number of bytes buffered before printing
cols	number of columns in a line
it	tabs initially every # spaces
lines	number of lines on a page
orc	horizontal resolution in units per character
orhi	horizontal resolution in units per inch
orl	vertical resolution in units per line
orvi	vertical resolution in units per inch
cps	average print rate in characters per second
Strings:	
cr	carriage return
cpi	change number of characters per inch
lpi	change number of lines per inch
chr	change horizontal resolution
cvr	change vertical resolution
csnm	list of character set names
mgc	clear all margins (top, bottom and sides)
hpa	horizontal position absolute
cud1	down one line
cuf1	carriage right
swidm	enable double wide printing
rwidm	disable double wide printing
ff	page eject
is1	printer initialization string
is2	printer initialization string
is3	printer initialization string
if	name of initialization file

(Continued on next page)

Table 6-8 terminfo entry definitions for printers
(Continued)

terminfo item	Meaning
iprog	pathname of initializing program
cud	move carriage down # lines
cuf	move carriage right # columns
rep	repeat a character # times
vpa	vertical position absolute
scs	select character set
smgb	set bottom margin at current line
smgbp	set bottom margin
smgl	set left margin at current column
smglp	set left margin
smgr	set right margin at current column
smgrp	set right margin
smgt	set top margin at current line
smgtp	set top margin
scsd	start definition of a character set
ht	tab to next 8-space tab stop

Consult the manual page for the `terminfo(M)` file structure for details on how to construct a *terminfo* database entry for a new printer.

Once you make the new entry, you need to compile it into the database using the `tic` program. Just enter the following command:

```
tic filename
```

filename is the name of the file containing the *terminfo* entry you have crafted for the new printer.

NOTE The print service gains much efficiency by *caching* information from the *terminfo* database. If you add or delete *terminfo* entries or change the values that govern pitch settings, page width and length, or character sets, you should stop and restart the print service so it can read the new information.

How to write an interface program

If you have a printer that is not supported by simply adding an entry to the *terminfo* database, or if you have printing needs that are not supported by the standard interface program, you can furnish your own interface program. It is a good idea to start with the standard interface program (or one of the many others found in `/usr/spool/lp/model`) and change it to fit, rather than starting from scratch. The “standard” script is found under the name `/usr/spool/lp/model/standard`.

What does an interface program do?

Any interface program performs the following tasks:

- Initializes the printer port, if needed. The generic interface program uses the `stty` command to do this.
- Initializes the physical printer. The generic interface program uses the `terminfo` and the `TERM` shell variable to get the control sequences to do this.
- Prints a banner page, if needed.
- Prints the correct number of copies of the request content.

An interface program is not responsible for opening the printer port. This is done by the print service, which calls a dial-up printer if that is how the printer is connected. The printer port connection is given to the interface program as standard output, and the printer is set to be the controlling terminal for the interface program so that a hang-up of the port causes a `SIGHUP` signal to be sent to the interface program.

A customized interface program must not terminate the connection to the printer or in any fashion uninitialized the printer. This restriction allows the print service to use the interface program only for preparing the printer and printer port, while the printing of content is done elsewhere, by the print service, for example, for preprinted form alignment patterns.

How an interface program is used

When the print service routes an output request to a printer, the interface program for the printer is invoked as follows:

```
/usr/spool/lp/admins/lp/interfaces/printer id user title copies options file1 file2 ...
```

Arguments for the interface program are:

printer	Printer name. This is the name given to the interface program itself.
id	Request id returned by lp.
user	Login name of user who made the request.
title	Optional title specified by the user.
copies	Number of copies requested by user.
options	List of options separated by blanks, specified by user or set by the print service.
file	Full pathname of a file to be printed.

When the interface program is invoked, its standard input comes from */dev/null*, its standard output is directed to the printer port, and its standard error output is directed to a file that will be given to the user who submitted the print request.

The standard interface recognizes the following values in the list in *options*:

nobanner This option is used to skip the printing of a banner page; without it, a banner page is printed.

nofilebreak

This option is used to skip page breaks between separate data files; without it, a page break is made between each file in the content of a print request.

cpi=decimal-number1

lpi=decimal-number2

These options say to print with *decimal-number1* characters per inch and *decimal-number2* lines per inch, respectively. The standard interface program extracts from the *terminfo* database the control sequences needed to initialize the printer to handle the character and line pitches.

The words *pica*, *elite*, and *compressed* are acceptable replacements for the *decimal-number1* and are synonyms for 10 characters per inch, 12 characters per inch, and as many characters per inch as possible.

length=decimal-number1

width=decimal-number2

These options specify the length and width, respectively, of the pages to be printed. The standard interface program extracts from the *terminfo* database the control sequences needed to initialize the printer to handle the page length and page width.

stty='stty-option-list'

The *stty-option-list* is applied after a default *list* as arguments to the *stty* command. The default list is used to establish a default port configuration; the additional list given to the interface program is used to change the configuration as needed.

The above options are either specified by the user when issuing a print request or by the print service from defaults given by the administrator for the printer (*cpi*, *lpi*, *length*, *width*, *stty*) or for the preprinted form used in the request (*cpi*, *lpi*, *length*, *width*).

Additional printer configuration information is passed to the interface program in shell variables:

TERM=printer-type

This shell variable specifies the type of printer. The value is used as a key for getting printer capability information from the extended *terminfo* database.

FILTER='pipeline'

This shell variable specifies the filter to use to send the request content to the printer; the filter is given control of the printer.

CHARSET=character-set

This shell variable specifies the character set to be used when printing the content of a print request. The standard interface program extracts from the *terminfo* database the control sequences needed to select the character set.

A customized interface program should either ignore these options and shell variables or should recognize them and treat them in a consistent manner.

Customizing the interface program

Make sure that the custom interface program sets the proper *stty* modes (terminal characteristics such as baud rate or output options). The standard interface program does this, and you can follow suit. Look for the section that begins with the shell comment:

```
## Initialize the printer port
```

Follow the code used in the standard interface program. It sets both the default modes and the adjusted modes given by the print service or the user with a line like the following:

```
stty mode options 0<&1
```

This command line takes the standard input for the *stty* command from the printer port. An example of an *stty* command line that sets the baud rate at 1200 and sets some of the option modes is shown here:

```
stty -parenb -parodd 1200 cs8 cread clocal ixon 0<&1
```

One printer port characteristic not set by the standard interface program is hardware flow control. This is set depending on your computer hardware. The code for the standard interface program suggests where this and other printer port characteristics can be set. Look for the section that begins with the shell comment:

```
# Here you may want to add other port initialization code.
```


Because different printers have different numbers of columns, make sure the header and trailer for your interface program correspond to your printer. The standard interface program prints a banner that fits on an 80-column page (except for the user's title which may be longer). Look for the section in the code for the standard interface program that begins with the shell comment:

```
## Print the banner page
```

The custom interface program should print all user-related error messages on the standard output or on the standard error. The messages sent to the standard error are mailed to the user; the messages printed on the standard output end up on the printed page, where they can be read by the user when they pick up the output.

When printing is complete, your interface program should exit with a code that tells the status of the print job. Exit codes are interpreted by the print service as shown in Table 6-9.

Table 6-9 Exit codes

Code	Meaning to the print service
0	The print request completed successfully. If a printer fault occurred, it was cleared.
1 to 127	A problem was encountered in printing this particular request (for example, too many non-printable characters or the request exceeds the printer capabilities). This problem does not affect future print requests. The print service notifies the person who submitted the request (via <code>write(C)</code> or <code>mail(C)</code>) that there was an error in printing it. If a printer fault occurred, it was cleared.
128	Reserved for internal use by the print service. Interface programs must not exit with this code.
129	A printer fault was encountered in printing the request. This problem affects future print requests. If the fault recovery for the printer directs the print service to wait for the administrator to fix the problem, it disables the printer. If the fault recovery is to continue printing, the print service does not disable the printer but tries printing again in a few minutes.
> 129	These codes are reserved for internal use by the print service. Interface programs must not exit with codes in this range.

As the table shows, one way of alerting the administrator to a printer fault is to exit with a code of 129. Unfortunately, if the interface program exits, the print service has no choice but to reprint the request from the beginning when the fault is cleared. Another way of getting an alert to the administrator but without requiring reprinting the entire request, is to have the interface

program send a fault message to the print service but wait for the fault to clear. When the fault clears, the interface program can resume printing the user's file. When finished printing, it can give a zero exit code as if the fault never occurred. An added advantage is that the interface program can detect when the fault is cleared automatically so that the administrator does not have to enable the printer.

Fault messages can be sent to the print service using the `lp.tell` program. This is referenced using the `LPTELL` shell variable in the standard interface code. The program takes its standard input and sends it to the print service, where it is put into the message that alerts the administrator to the printer fault. If its standard input is empty, `lp.tell` does not initiate an alert. Examine the standard interface code immediately after these comments for an example of how the `lp.tell (LPTELL)` program is used:

```
# Here's where we set up the $LPTELL program to capture
# fault messages.

# Here's where we print the file.
```

With the special exit code 129 or the `lp.tell` program, there is no longer the need for the interface program to disable the printer itself. Your interface program can disable the printer directly, but doing so overrides the fault alerting mechanism. Alerts are sent only if the print service detects the printer has faulted, and the special exit code and the `lp.tell` program are its main detection tools.

If the print service has to interrupt the printing of a file at any time, it kills the interface program with a signal 15 (see `signal(S)` and `kill(C)` for more information).

If the interface program dies from receipt of any other signal, the print service assumes that future print requests are not affected and continues to use the printer. The print service notifies the person who submitted the request that it did not finish successfully.

The signals `SIGHUP`, `SIGINT`, `SIGQUIT`, and `SIGPIPE` (trap numbers 1, 2, 3, and 13) start out being ignored when the interface is invoked. The standard interface changes this to trap these signals at appropriate times. The standard interface considers receipt of these signals as meaning the printer has a problem and issues a fault. This is the program the print service uses to manage the printer each time a file is printed. It has four main tasks:

- to initialize the printer port (the connection between the computer and the printer),
- to initialize the printer (restore it to a normal state in case a previously printed file has left it in an unusual state) and set the character pitch, line pitch, page size, and character set requested by the user,

6. Using printers

- to print a banner page, and
- to run a filter to print the file.

How to add an interface program

If you do not choose an interface program, the standard one provided with the print service is used. This should be sufficient for most of your printing needs. If you prefer, however, you can change it to suit your needs or completely rewrite your own interface program, and then specify it when you add a new printer.

If you plan to use the standard interface program, you need not specify it when adding a printer. However, if you use a different interface program, you can either refer to it by its full pathname or by another printer using the same interface program.

To identify a customized interface program by name, give the printer name and the pathname of the interface program as follows:

```
/usr/lib/lpadmin -p printername -i pathname
```

To identify a customized interface program by reference to another printer, give the printer names as follows:

```
/usr/lib/lpadmin -p printername1 -e printername2
```

printername1 should be replaced with the name of the printer you are adding; *printername2* should be replaced with the name of the printer already added that is using the customized interface program.

To identify an interface program by reference to a model interface program, give the printer name and model name as follows:

```
/usr/lib/lpadmin -p printername -m modelname
```

This takes the interface program from the file */usr/spool/lp/model/modelname*.

NOTE If *printername* is a new printer to be added, the `lpadmin` command line must include the `-u` (dial-out) or `-v` (device) option. The printer must then be enabled and accepted.

Configuring a network printer

To configure the spooler to print to a printer on another machine accessed via network (such as TCP/IP or UUCP) you must set up the printer as follows:

1. Choose Printers ⇨ Configure ⇨ Add from `sysadmsh`. Give the printer a name, and choose the existing interface script *network* (or *network.ps* for a PostScript-capable printer). The Connection should be “Direct” and the Device should be “Hardwired.” If you are adding a printer on a TCP/IP

network, use `/dev/null` for the device name, otherwise, this should be left blank.

2. Create a file called `/usr/spool/lp/remote`. This file should contain the following line:

```
name: <Tab> command
```

where *name* is the name of the printer on this machine and *command* is the command which will transfer the file to be printed to the remote system. For example, if the name you gave the printer is *remprint* and the name of the remote machine to which this printer is connected is *wally* and the name of the printer on machine *wally* is called *beaver* then the line would read:

For UUCP:

```
remprint:      uux - wally! /usr/bin/lpr -dbeaver
```

For TCP/IP:

```
remprint:      /usr/spool/lp/bin/rlpcmd wally lp -dbeaver
```

3. The file `/usr/spool/lp/remote` should have owner and group *lp*, and be publicly readable. To set this, issue the following commands:

```
chown lp /usr/spool/lp/remote
chgrp lp /usr/spool/lp/remote
chmod 444 /usr/spool/lp/remote
```

4. For a TCP/IP network, a user need not have an account on the print server; all that is necessary is consistency among the *lp* accounts on the network. The *.rhosts* file should contain entries in the following format:

```
client  lp
```

All print jobs are submitted under the name *lp*.

NOTE If you have configured a remote printer over UUCP, some information about dialup printers also applies to UUCP. See the next section about dialup printers.

Configuring a dialup printer

To set up a dialup printer, you must do the following:

1. Because the *cu* program accesses a printer in the same way the print service does, you should set up the files as though preparing access to the printer for *cu*. The *cu* command is not used to access printers but can serve as a yardstick when setting up files: if *cu* can access a printer, the print service will be able to access it, too. (See the "Using modems" chapter in *Administering Networking Services* for details about setting up a modem connection.)

6. Using printers

2. Make certain the following line is present in */etc/default/lpd*:
`DIALUPPRINTER=YES`
If it is missing or the entry reads "NO", change it.
3. Choose Printers ⇨ Configure ⇨ Add from `sysadmsh`. Complete the fields as follows:

Name:	dialup (or any name)
Interface:	dumb
Dial-up information:	the phone number of the remote system
Device:	hardwired

Press (Ctrl)x when you are finished.
4. Enter the following command to ensure failure info is mailed to *root*:
`/usr/lib/lpadmin -p dialup -A mail`
5. You must then prepare the print system to accept jobs and enable the device:
`accept dialup`
`enable dialup`
6. You should then be able to submit job with the following command:
`lp -ddialup filename`

The following applies to dialup and UUCP network printers: If the printer or port is busy, the print service will automatically retry later. This retry rate is 10 minutes if the printer is busy and 20 minutes if the port is busy. The rate is not adjustable. However, you can force an immediate retry by issuing an **enable** command for the printer. If the port or printer is likely to be busy for an extended period, you should issue a **disable** command.

The `lpstat -p` command reports the reason for a failed dial attempt. Also, if you are alerted to a dialing fault the alert message will give the reason for the fault. (Refer to "Fault alerting" (page 129) for more information on fault alerting.) These messages are identical to the error messages produced by the UUCP system for similar problems. Refer to the section on UUCP status error messages in the *Administering Networking Services* for an explanation of the reasons for failure.

Using a printer without the spooler

If you use a printer without the spooler, any `stty` settings you have specified for use with that printer do not stay in effect. The spooler opens the file and then runs the `stty` commands as specified in the printer interface script. To use a printer without the spooler, follow the instructions in this section.

While logged in as `root`, give the following commands or insert them into the initialization file `/etc/rc2.d/S80lp` before the line that calls `/usr/lib/lpsched`. The first command is for serial printers, and the second for parallel printers:

```
(stty baud ixon ixoff -ixany ; cat > /dev/null) < /dev/tty &
(stty onlcr ; while : ; do sleep 3600 ; done) < /dev/lp &
```

where *baud* is the baud rate of the printer, *ttyn* or *lpn* is the serial or parallel device name, respectively. This command sets the `stty` options and holds the port open for use without the spooler. If you ever need to enable the port, make sure you kill this process first.

NOTE With certain multiport cards, it is necessary to add a `sleep` command after the initialization program supplied with the card, *initprogram*, followed by the `stty holdopen` command:

```
initprogram &
sleep 3
```

Creating an init device file

The standard parallel printer devices (`/dev/lp`, `/dev/lp0`, `/dev/lp1`, and `/dev/lp2`) send a printer initialization string (`init`) the first time the device is opened after the system starts up. This is done on the first open only, so printers with large text buffers are not flushed by the sending of another file.

Some parallel printers require initialization every time a file is received for printing. Others require an `init` if the printer is turned off and back on again (for example, after changing paper or ribbons). The symptom of this situation is that the printer works fine until it is turned off and then back on.

If you need to initialize the printer more often than the standard devices provide, you can create an additional device file for the parallel port in use. This init device file can be used when necessary to initialize the printer.

1. Log in as super user.
2. Determine which device is the parallel port you are using. This example assumes the device is the main parallel port (`/dev/lp0`).

3. Run the command:

```
cd / ; fixperm -csdLPR /usr/lib/mkdev/perms/LPR
```

to correct/initialize file permissions for the printer devices file */usr/lib/mkdev/perms/LPR*.

4. Use the **sysadmsh** selection:

Printers ⇔ Configure ⇔ Modify

to associate one of the parallel init devices (*lp0i*, *lp1i*, *lp2i*) with the printer.

If your printer needs an init when it is turned off and on, use the following command line after the printer is turned on. Initialize the printer before the first file is sent to the printer (this example assumes the main parallel port):

```
> /dev/lp0i
```

If your printer needs an init every time a file is sent (and it does not have a large internal text buffer), you can use the */dev/lp0i* device all the time.

The **lp(C)** command will then send an init every time a file is sent to the printer.

Chapter 7

Using floppy disks and tape drives

An important part of any computer system is the ability to offload files and restore them when needed. There are several types of media used to store and recall files. Among these are floppy disks and magnetic tape devices. This chapter explains how to use storage media with your system. Your system should come with at least a floppy disk drive already installed and ready to run. For information on configuring tape drives, consult the *Hardware Configuration Guide*.

Using tape drives

A tape drive is a mass storage device that uses magnetic tape cartridges to store data. A tape can hold many times the data that can be stored on floppies, making it much more useful for large backup operations.

The drives that are supported are listed in the SCO list of compatible hardware. For hardware-specific information, refer to the hardware manual for your drive, the manufacturer of the drive, and the tape(HW) manual page.

Tape device files

After the drive is configured, a series of device files are created to access the tape drive. Some are linked to the basic tape device file, */dev/rct0*. Others include the “no-rewind” device, which does not rewind the tape after access, and the “no-unload” device, which does not eject the tape after use. Table 7-1 lists the prefixes used in QIC-02 and SCSI tape device names.

Table 7-1 Tape device prefixes

Prefix	Meaning	Example
e	ECC device (QIC-02 only)	/dev/erct0
rh	high density (9-track only)	/dev/rhStp0
nr	no rewind, no unload	/dev/nrStp0
nur	rewind, no unload	/dev/nurStp0
ur	rewind, unload	/dev/urStp0
r	rewind on close	/dev/rStp0, /dev/rct0
x	override/control	/dev/xStp0, /dev/xct0, /dev/xft0, /dev/xctmini
n	no rewind	/dev/nrct0

Table 7-2 lists the main devices created for tape drives and the default device to which they are linked.

Table 7-2 Tape device files

Tape device	Device file	Default device
QIC-02	/dev/rct0	/dev/rct0
Mini (irwin)	/dev/rmc0	/dev/rctmini
QIC-40/80	/dev/rft0	/dev/rctmini
SCSI cartridge	/dev/urStp0	/dev/rct0
DAT/Exabyte	/dev/nurStp0*	/dev/rct0
9-track	/dev/nurStp0	/dev/rct0
Compaq	/dev/rmt/cst2*	/dev/rct0

* Some drives have additional device files that depend on brand and model. For example, certain DAT drives support tape partitioning; see `dat(HW)` and `tape(HW)`.

In addition, some devices are linked to equivalent standard UNIX devices in the `/dev/rmt/*` hierarchy.

Changing the default tape drive

If you install more than one tape drive, you are prompted by `mkdev tape` for the drive you want linked to the default tape device files. For example, if you have both a SCSI cartridge tape drive and an 8-mm tape drive installed, only one of them can be linked to the default device `/dev/rct0`. If you decide later to change the drive linked to the default device, you can do so with option 3 of the `mkdev tape` menu.

Setting the default tar(C) device

After you install your tape drive, you must enter the correct size setting in the `/etc/default/tar` file. When you edit the file, you see several entries for various default devices. Figure 7-1 shows the `/etc/default/tar` file provided with the 96tpi distribution. Note that the sizes for `/dev/rct0` and `/dev/rctmini` are 0, meaning there is no size associated with this device. This is acceptable if you never use an entire tape. However, if you exceed the tape's capacity, `tar` cannot create a multi-volume archive properly. If you plan to create archives that span tape volumes, you must add the proper tape size as described in this section.

```
#device      block  size  tape
archive0=/dev/rfd048ds9  18    360  n
archive1=/dev/rfd148ds9  18    360  n
archive2=/dev/rfd096ds15 10   1200  n
archive3=/dev/rfd196ds15 10   1200  n
archive4=/dev/rfd096ds9  18    720  n
archive5=/dev/rfd196ds9  18    720  n
archive6=/dev/rfd0135ds18 18   1440  n
archive7=/dev/rfd1135ds18 18   1440  n
archive8=/dev/rct0       20     0   y
archive9=/dev/rctmini    20     0   y
# The default device...
archive=/dev/rfd096ds15  10   1200  n
```

Figure 7-1 `/etc/default/tar` file

QIC cartridge drives

If a QIC-02 drive is your default tape drive, it uses the `/dev/rct0` entry. The cartridge sizes are indicated in Table 7-3. You can edit `/etc/default/tar` and set the appropriate size, or create multiple entries to accommodate different tape sizes.

Table 7-3 QIC cartridge sizes

Length in feet	Entry in "Size" field	Capacity in megabytes
300	30000	30
450	45000	45
600	60000	60
1200	120000	120
1500	150000	150
2500	250000	250
5250	525000	525

Mini-cartridge (Irwin) drives

Find the entry in your `/etc/default/tar` file for `/dev/rctmini`. In Figure 7-1, this is `archive9`. The correct size for your `rctmini` device varies with the size of the tape you use. See Table 7-4.

Table 7-4 Mini-cartridge sizes

Tape size in megabytes	Actual capacity in megabytes	Entry in "Size" field
40	35	35000
80	72	72000
120	118	118000

The actual size for Irwin tapes is smaller because some storage is reserved for error correction code (ECC) data.

DAT and 8-mm (Exabyte) drives

If you wish to use the default file with a DAT or 8-mm drive and it is not the default tape device, you must change one of the entries to use the `/dev/rStp` device:

```
archive9=/dev/rStpn    20    size    y
```

Replace `n` with the number of your drive and `size` with the size (in Kbytes) of the cartridges you are using.

Archiving files on tape

You use a tape drive much like a floppy, but the volume of data stored is much greater. Tapes are much better for storing (backing up) entire filesystems. The `tar(C)` command is the recommended archive program for users and is best used for general archiving or transporting of files. Other programs such as `cpio(C)` are meant for system administrators making copies of entire filesystems. Consult "Backing up and restoring your system" in the *Installation and Upgrade Guide*.

The `cpio(C)` command is a general-purpose archive program that uses a different format than `tar`. The `dd(C)` program transfers or converts archives of unusual format; the input and output format can be specified on the command line.

The tar command

The `tar` command is useful for making a backup copy of entire directories. The command has the syntax:

```
tar cvf device file files
```

The *devicefile* is the filename that corresponds to the cartridge drive. *files* are the names of the files or directories to be copied. For example, to copy all the files in the directory */u/bogart* to the cartridge drive */dev/rct0*, enter:

```
tar cvf /dev/rct0 /u/bogart
```

Δ **sysadmsh** users select: Media ⇌ Archive

WARNING **tar** will not accept pathnames longer than 100 characters. It will fail with the following error message:

```
tar: filename : pathname too long
```

cpio(C) accepts pathnames of 1024 characters.

Alternatively, you can use the shorter */etc/default/tar* entry (**archive8**) shown in Figure 7-2 rather than entering the full device name:

```
tar cv8 /u/bogart
```

To restore files stored on tape, insert the cartridge containing the files or directories you wish to restore and enter the following command:

```
tar xvf devicefile
```

Δ **sysadmsh** users select: Media ⇌ Extract

Again, you can also use the shorter form:

```
tar xv8
```

NOTE When creating a **tar** archive, you need to be conscious of absolute and relative pathnames. If you specify an *absolute* pathname when creating an archive (for example: **tar cv2 /u/wanda/tempfile**), then it will be extracted with that same pathname (and thus appear below the current directory). If you do not wish this to occur, use *relative* pathnames when creating an archive (for example: **tar cv2 tempfile**).

You can list (or verify) a **tar** archive by asking for a listing with the **t** option:

```
tar tv2
```

Tape drive maintenance

The **tape(C)** utility performs various tape maintenance operations on all tape drives. **tape** sends commands and receives status from the tape drive. The basic form of the command is:

```
tape command [ devicefile ]
```

For example, to rewind a cartridge tape device, enter:

```
tape rewind
```

When invoked without a device, the default device (in */etc/default/tape*) is used.

Other commands include the following:

- erase** erases tape cartridge and re-tensions
- reset** resets tape controller and tape drive, clears error conditions, and returns tape subsystem to power-up state
- reten** re-tensions tape cartridge. Should be used periodically to remedy slack tape problems that generate an unusually large number of tape errors or, in extreme cases, actually tangle the tape in the drive.

Refer to the `tape(C)` manual page for more information on these and other commands.

After certain tape operations are executed, the system returns a prompt before the tape controller has finished its operation. If you enter another tape command too quickly, the message "device busy" is displayed until the tape device is finished with its previous operation.

You should clean the tape drive heads and re-tension cartridges to keep it operating error-free.

Irwin drive commands

There are certain commands that are specific to Irwin drives:

- info** prints information about the inserted cartridge, including whether or not it is formatted, what type of format, and write protect status
- capacity or kapacity** displays the capacity of the inserted cartridge in 512- and 1024-byte blocks, respectively

Tape driver error correction code (ECC) support

Tape ECC is supported on QIC-02, mini-cartridge (Irwin), and QIC-40/80 drives. ECC is not available on SCSI tape drives.

QIC-02 ECC

The QIC-02 ECC tape device node, `/dev/erct0`, is automatically created when you run `mkdev tape`. To use ECC on the QIC-02, you must read and write from this device, not the normal `/dev/rct0`. With tape drives that support cartridges larger than 60 Mbytes, it is a good idea to edit the `/etc/default/tar` file and substitute `/dev/erct0` for the normal tape device.

The error recovery scheme is 2/64, which means that two 512-byte blocks out of every 64 blocks can go bad and the driver corrects them. The probability of error with ECC is $1:10^{14}$. Standard drives have an error probability of $1:10^9$.

Be sure and label tapes that are created with the ECC device; these tapes cannot be read by standard devices. In addition, if transporting data from one machine to another, it is advisable to use the ECC device only if the target machine supports the ECC scheme.

Mini-cartridge (Irwin)

ECC encoding and decoding is automatic on the mini-cartridge (Irwin) 80-Mbyte drives; no ECC device is necessary. For every 29K written to the tape, 3K of ECC data is written. On smaller capacity mini-cartridge drives, for every 16K written to the tape, 2K of ECC data is written.

Using floppy disks

Floppy disks are the most convenient form of storage media. Depending on your floppy disk drive, you may be able to store from 360 Kbytes to 1.4 Mbytes on a single disk. Floppy disks can be used for simple data storage in `tar`, `cpio`, or `dd` formats or you can make a mountable filesystem on a floppy disk. The following sections explain how to use floppies for data storage and as extra filesystem space.

Formatting floppy disks

Floppy disks must be formatted before they can be used. The command to format a floppy disk is:

format /dev/floppy-device

Δ `sysadmsh` users select: Media ⇌ Format

The floppy device you specify in the command relates to the type of disk drive and floppy you are using. For example, if you have a high-density 5.25-inch floppy disk drive, you can use it in high-density mode (96 tracks per inch) or in low-density mode (48 tpi). If you have high-density floppies to use with your drive, the floppy device to specify is:

/dev/rfd096

In this example, “rfd” indicates the raw floppy device, “0” indicates that this is the primary floppy drive, and “96” indicates high-density mode.

NOTE You must always specify the raw device; you cannot format the block device.

Similarly, if you wish to use low-density floppies and the low-density mode of the floppy drive, the device name is:

/dev/rfd048

In this example, “48” indicates the low-density mode of floppy drive “0”.

etc/default/format file

You can also define a default format device by adding an entry to the file */etc/default/format*. For example:

```
DEVICE=/dev/rfd096ds15
```

After adding the above line, you no longer have to specify the device name. In addition, it is possible to define that all floppies be verified, which confirms that the data on the floppy is readable. (This can also be specified on the command line with the *-v* option.) Automatic verification can be specified by the following entry:

```
VERIFY=Y
```

If this entry is placed in */etc/default/format*, all floppies formatted with the *format* command are verified. (To override verification, use the *-n* option on the command line.)

Refer to the *format(C)* manual page for more details.

Copying floppy disks

To ensure against the loss of data stored on floppy disks, any user can use the *diskcp(C)* command, or the *dd(C)* command to make copies of floppy disks on new, formatted disks.

diskcp makes use of *dd* and provides a simple interface to that program. *dd* is very powerful, and you can use it to perform many different kinds of copying.

If you need to copy one type of floppy to another (for example: 5¼ to 3½) you must use *dd* directly.

You must copy information onto formatted disks. If you format floppies, you can use them over again without reformatting.

NOTE If you have disks that were formatted under another operating system, you must reformat them before you can use them to make copies of UNIX disks. Be aware that floppies formatted under some operating systems cannot be used under other operating systems, even with reformatting.

You can use the *format* command to format floppies. Refer to “Formatting floppy disks” (page 173) for more information. The *diskcp* command can also format floppies for you.

To copy a floppy disk using **diskcp**, do the following:

Δ **sysadmsh** users select: Media ⇔ Duplicate

1. Insert the disk you want to copy, known as the *source* floppy, in drive 0, your primary floppy drive.
2. Insert another floppy in the other drive. This floppy is known as the *target* disk. Note that any information already on the target disk is destroyed.
If you have only one disk drive, leave the source floppy in the drive. **diskcp** prompts you to remove the source disk at the correct time.
3. To format the floppy disk before the image is copied, enter the command:

diskcp -f

If your computer has two identical floppy drives, enter the following command to copy the image directly on the target floppy:

diskcp -d

If you do not need to format the target floppy, simply enter:

diskcp

4. Follow the instructions as they appear on your screen. Note that, with a single drive system, you are prompted to remove the source disk and insert the target disk.

To copy a disk using **dd**, follow these steps:

1. Insert the disk to be copied into floppy drive 0.
2. Insert a formatted disk into drive 1. If necessary, you can format a disk with the **format** command. Refer to "Formatting floppy disks" (page 173) for more information.
3. Enter:

dd if=/dev/fd0 of=/dev/fd1 count=blkcount

The **blkcount** is the number of blocks on the disk to be copied. If you do not know this number, leave the **count=blkcount** section out of the command.

This command copies the first disk to the second, then displays a record of the number of blocks copied.

Using floppies for file storage

To use a floppy for simple file storage, first make sure that the floppy is formatted. Then, place the floppy in the floppy drive. You can use any of the standard UNIX file archiving utilities with floppy disks. These include **tar**, **cpio**, or **dd** formats.

tar is recommended for most file-archiving tasks. See "The **tar** command" (page 170) for details. For information on **cpio**, **dd**, and **backup** formats, see the associated manual pages.

Making filesystems on floppy disks

You can make a filesystem on a floppy disk similar to how you make one on a hard disk. Filesystems on floppy disks are portable and can be mounted on any UNIX system. A special directory called **/mnt** is used for mounting filesystems that do not have a specified mounting point. Note that for system security, you must be logged in as **root** to use floppy filesystems, and only **root** can mount a floppy filesystem.

To make a portable filesystem on a floppy disk, use the following procedure:

1. Log in as **root** and enter the command:
mkdev fd
Δ sysadmsh users select: Filesystems ⇨ Floppy
2. You see the following menu:

```
Floppy Disk Filesystem Creation Program
Choices for type of floppy filesystem.
1. 48tpi, double sided, 9 sectors per track
2. 96tpi, double sided, 15 sectors per track
3. 135tpi, double sided, 9 sectors per track
4. 135tpi, double sided, 18 sectors per track
Enter an option or q to quit:
```

Enter the number of the disk type desired and press (Return).

3. If you have more than one floppy drive you see the following prompt:

```
Do you want to use floppy drive 0 or floppy drive 1
```

Enter the number of the drive required and press (Return).

4. Next you see:

```

Choices for contents of floppy filesystem.
1. Filesystem
2. Bootable only          (96ds15 and 135ds18 only)
3. Root filesystem only (96ds15 and 135ds18 only)

Enter an option or enter q to quit:

```

Enter 1 and press (Return).

5. You see the following prompt:

```

Insert a type floppy into drive 0.
Press Return to continue or enter q to quit:

```

Press (Return).

6. The following prompt is displayed:

```

Would you like to format the floppy first? (y/n)

```

If you have already formatted the floppy, enter "n" and the filesystem is immediately created. If the floppy has not yet been formatted, enter "y" and you see:

```

formatting /dev/type
track 00 head 0

```

The track and head numbers count up as the floppy is formatted. (If */etc/default/format* contains **VERIFY=Y**, the format is also verified after formatting.)

7. Next you see a prompt for filesystem type:

```

Do you want to use the default file system type AFS (y/n)?

```

If you respond "n", you are asked to supply the type:

```

Please enter a file system type from <AFS, EAFS, S51K, XENIX>
or enter q to quit:

```

7. Using floppy disks and tape drives

8. Enter the filesystem type and press <Return>. When the process is complete, you see:

```
Filesystem creation complete.
```

9. Next you see this menu again:

```
Choices for contents of floppy filesystem.
1. Filesystem
2. Bootable only      (96ds15 and 135ds18 only)
3. Root filesystem only (96ds15 and 135ds18 only)
Enter an option or enter q to quit:
```

Now enter "q" and press <Return> to quit. Your floppy now contains a filesystem.

Mounting a floppy filesystem

To use a floppy filesystem, you must mount it on your system. For example, a 96-tpi floppy would use the following command:

```
mount /dev/fd096 /mnt
```

Δ sysadmsh users select: Filesystems ⇌ Mount

Note that you use the floppy device *fd096* and not *rfd096*. When you mount a floppy filesystem, you must use the name without the preceding "r". As another example, if you choose to mount a filesystem on a 48-tpi disk, use the following command:

```
mount /dev/fd048 /mnt
```

When you give the **mount** command, the shell should return a prompt. This indicates that the filesystem was successfully mounted. You can now use the **cd** command to move into the filesystem and create files there as you would on any other filesystem:

```
cd /dev/fd048
```

When you are done and you wish to remove the floppy, be sure and **cd** out of the floppy filesystem and give the following command:

```
umount /dev/fd048
```

Δ sysadmsh users select: Filesystems ⇌ Unmount

Your filesystem is immediately unmounted. Your files are contained on the floppy and can be stored or transported easily.

Chapter 8

Using MS-DOS and other DOS operating systems

Many users received MS-DOS, or other closely compatible DOS operating systems with their computer. This chapter explains how you can still use DOS utilities, files, and applications after you install the UNIX system. You can even access DOS files and directories on your UNIX system, or mount DOS file-systems and access the files directly. The UNIX system provides this facility so that you do not need to throw away your investment in DOS software, or buy another computer just to run a UNIX system.

NOTE This chapter is only concerned with accessing a DOS partition using facilities under the UNIX system. For information on accessing and executing DOS files under DOS Services, consult the DOS Services documentation.

Several programs make this coexistence possible. The `dos(C)` utilities allow access to DOS files on diskettes or on the DOS partition on the hard disk (provided the partition is unmounted). These utilities are discussed later in this chapter. The utility that partitions the disk is called `fdisk(ADM)` and is available in DOS and UNIX system versions. The next section explains how to use `fdisk` to create a DOS partition and a UNIX system partition on the same hard disk. Another section discusses installing a UNIX system partition on the hard disk along with DOS. There is also a section explaining various booting configurations, for users who mostly use the UNIX system and for users who mostly use DOS.

OS/2 coexistence

Although it may install successfully, OS/2 may not be bootable on your machine, regardless of whether a UNIX system partition is present or not; we cannot guarantee that OS/2 will work with your UNIX system. Refer to your computer's hardware documentation to determine if your machine is supposed to run OS/2. If you wish to use OS/2 and/or DOS on the same disk with your UNIX system, you must install them on the disk in the following order:

1. DOS Primary Partition
2. DOS Extended Partition
3. UNIX system
4. OS/2

NOTE If a second disk is installed you get a primary partition on the second disk and optionally an extended partition. You cannot boot DOS from the second extended partition.

There are no OS/2 tools available in UNIX System V (such as the DOS utilities described in this chapter). In addition, you must use `fdisk(ADM)` to switch to or from OS/2.

UNIX system `fdisk(ADM)` displays an OS/2 partition as DOS.

Partitioning the hard disk using `fdisk`

Each version of `fdisk` is documented in the respective operating system's manual. Unless otherwise noted, this chapter refers to the UNIX system version of `fdisk(ADM)`.

`fdisk` is interactive, and uses a menu to display your options. Here is the main `fdisk` menu:

- ```
1. Display Partition Table
2. Use Entire Disk For UNIX
3. Use Rest of Disk for UNIX
4. Create UNIX Partition
5. Activate Partition
6. Delete Partition
```

```
Enter your choice or 'q' to quit:
```

You can do the following using *fdisk*:

- Set up separate areas (partitions) on your hard disk for different operating systems. The hard disk is divided into *tracks*. The number of tracks depends upon the size of the hard disk. A *partition* consists of a group of tracks. Each hard disk can accommodate four partitions.
- Specify one disk partition as “active”. This means that when you turn on (boot) your computer, the operating system installed in the active partition will start running. The UNIX system partition must be active when you intend to use your UNIX system.
- Specify the number of tracks assigned to each partition. The number of available tracks will vary according to the size of your hard disk. Consult your *Installation and Upgrade Guide* for the recommended UNIX system partition size. The size of the UNIX system partition also depends on the number of software packages you want to install. You can install the UNIX system in this space, and have the rest of the space for user files and other software packages. Refer to the *custom(ADM)* manual page for information on how to install and remove software.
- Specify precisely where the partition begins and ends. *fdisk* will not allow you to construct overlapping partitions. You do not need to install your UNIX system in the first partition.

You should start your DOS partition at the beginning of the disk, starting at cylinder 0 or cylinder 1.

If you install a UNIX system partition on the same disk after the DOS, or extended DOS partition, start the UNIX system partition at the beginning of the next cylinder on the disk. To find the beginning of the next cylinder, note the ending track number of your DOS partition and start the UNIX system partition on the next track number that is a multiple of the number of heads on your hard disk. For example, if you have five heads on your hard disk and your DOS partition ends at track 103, start your UNIX system partition at track 105.

When you are running your UNIX system, the device name of the UNIX system partition is */dev/hd0a*. For more information about hard disk device names, see the *hd(HW)* manual page.

## *Displaying current partition table*

---

One option of *fdisk* tabulates the current state of the partitions (the Display Partition Table option). This option lists, for each partition, whether the partition is active, the first track, the last track, the number of tracks used, and the associated operating system. If you enter the Display Partition Table option

and press (Return) to see the partition table, the result will be similar to Example 8-1:

### Example 8-1 Sample fdisk table

Current Hard Disk Drive: /dev/rhd00

| Partition | Status   | Type      | Start | End  | Size |
|-----------|----------|-----------|-------|------|------|
| 1         | Inactive | DOS       | 5     | 398  | 393  |
| 2         | Inactive | DOS (ext) | 400   | 1219 | 819  |
| 3         | Active   | UNIX      | 1220  | 2220 | 1000 |

Total disk size: 2229 tracks (9 reserved for masterboot and diagnostics)

## Switching operating systems

There are three ways to switch to the primary DOS partition once you have set up separate DOS and UNIX system partitions:

- enter **dos** at the boot prompt;
- use a floppy diskette that contains the files necessary to boot the DOS operating system, or
- use **fdisk** to change the current active partition.

When you use the boot prompt or a floppy to boot DOS, the UNIX system partition remains active even though you have switched operating systems. When you use **fdisk**, the UNIX system partition is inactive until you switch back to it.

To use the boot prompt method, enter:

**dos**

at the boot prompt:

```
SCO System V/386
```

```
Boot
: dos
```

**NOTE** The system boots from the first DOS partition found.

To use a floppy diskette to boot DOS, follow this procedure:

1. Make sure all users are logged off the system.

2. Run **shutdown(ADM)** to shut down the UNIX system. This command makes sure all users know the system is being shut down, terminates all processes, then halts the system.
3. Once the UNIX system has shut down, insert the bootable DOS diskette into the primary (boot) drive.
4. Boot DOS.
5. To get back to the UNIX system partition, remove any disks from the floppy drive(s) and press **(Ctrl)(Alt)(Del)**, or the reset key, or turn the computer off, then on. Because the UNIX partition is still active, your UNIX system boots.

Remember that if you have an active UNIX system partition and boot DOS from a floppy you can transfer to C: to work with the DOS files.

The other way to change operating systems is to run **fdisk** and change the active partition from the UNIX system partition to DOS. Then, after you shut down the system (see the previous steps) DOS boots from the hard disk. From here you can switch operating systems from the DOS partitions. You do not need a bootable DOS floppy disk as long as DOS is loaded on the DOS partition of the hard disk.

To switch back to the UNIX system partition, run **fdisk** under DOS and make the UNIX partition active. To reboot the UNIX partition, press **(Ctrl)(Alt)(Del)**, or the reset key, or turn the computer off, then on.

Because the UNIX system partition must be active for it to operate, you cannot use a bootable floppy to boot the operating system. This second method is appropriate for an occasional change of the active operating system.

**Table 8-1 DOS hard disk devices**

| XENIX device convention | UNIX device convention          |
|-------------------------|---------------------------------|
| /dev/hd0d               | /dev/dsk/0sd (linked with 0sC)  |
| /dev/rhd0d              | /dev/rdsk/0sd (linked with 0sC) |
| /dev/hd1d               | /dev/dsk/1sd                    |
| /dev/rhd1d              | /dev/rdsk/1sd                   |

The hard disk device names in Table 8-1 are similar to */dev/hd0a* (the active disk partition) in that the disk driver determines which partition is the DOS partition and uses that as *hd0d* and *hd1d*. (You can use the XENIX or UNIX system device name conventions; they are equivalent.) This means that software that is running from the UNIX system partition and using the DOS partition does not need to know which partition is DOS (the disk driver determines that).



## Installing a UNIX system partition on a DOS system

---

If you wish to set up your UNIX system on a hard disk which previously contained only DOS, follow these steps:

1. Copy (back up) all the DOS files and directories on the hard disk onto floppies, or whatever backup media you wish to use.
2. Run `fdisk`, under DOS. If there is enough free space for a UNIX system partition on your hard disk (check your *Installation and Upgrade Guide*), skip to Step 4. Otherwise, delete the DOS partition, then recreate it, leaving enough room on the disk for your SCO Open Desktop/SCO Open Server distribution and any other software that you intend to install.
3. Return the DOS files from the backup media to the newly created DOS partition on the hard disk. Keep the backups in case there is an error of some kind, so you will not lose any data.
4. Turn off your computer.
5. Follow the installation procedure outlined in the *Installation and Upgrade Guide* to install your SCO Open Desktop/SCO Open Server distribution.

You will see a message warning that the contents of the hard disk will be destroyed. There is no cause for concern, because you have already backed up the DOS files and transferred them to the new DOS partition. The new partition being created will contain your UNIX system, and the installation process will only write information on the UNIX system partition.

6. During the installation procedure, `fdisk` is invoked to partition the hard disk. Use `fdisk` to assign a sufficiently large UNIX system partition.
7. Designate "UNIX" as the active operating system by choosing the "Activate Partition" option under `fdisk`.
8. Finish installing the UNIX system distribution.

**NOTE** UNIX `fdisk` displays DOS partitions as "DOS" while DOS `fdisk` displays UNIX system partitions as "Other". You can only create DOS partitions using DOS `fdisk`, and UNIX system partitions using UNIX system `fdisk`.

Be aware that DOS `fdisk` reports sizes in terms of cylinders, while UNIX `fdisk` reports sizes in terms of tracks. Check your hard disk manual for the number and size of cylinders on your hard disk.

## Using a UNIX system and DOS with two hard disks

---

Your computer always boots the operating system in the active partition on the first hard disk. The UNIX system must boot from the first hard disk. There are several ways to configure your system if you have two hard disks and want to boot DOS. Two ways are discussed here.

One configuration consists of designating the entire first disk as a UNIX system partition. You then use a DOS boot floppy to start DOS and specify:

```
A> D:
```

to switch to the DOS area on the second hard disk, where D: is the designation for the second hard disk. This strategy works for some versions of DOS. Early versions recognize only the first hard disk on the system.

**NOTE** If you devote a hard disk for use with DOS, the disk must already be configured under DOS. See the *Hardware Configuration Guide* for details regarding hard disk configuration.

Another method is to maintain a small DOS partition on the first hard disk. The DOS partition is designated the active partition. In this configuration, the computer always boots DOS. This requires changing the active partition to boot the UNIX system from the hard disk.

If you use the entire second disk for DOS, you need only run `mkdev hd` to create device files for the second disk if you plan to use the UNIX system DOS utilities (`doscp`, `dosls`, `doscat`, and so on). If you do not wish to use those utilities to access DOS files on the second hard disk, there is no need to run `mkdev hd`.

**NOTE** Be sure to make a backup copy of your boot floppies if you use them to boot your secondary operating system.

## Removing an operating system from the hard disk

---

You may find that you no longer need one of the operating systems installed on your hard disk. If you want to delete an operating system, use the appropriate version of `fdisk`. To delete a UNIX system partition, you must use the UNIX system version of `fdisk`. To delete a DOS partition, use `fdisk` under DOS. To delete an Extended DOS partition, you must delete all logical drives on that partition using `fdisk`. Deleting the partition removes the contents of that partition and leaves unallocated space.

You can then reallocate that space by either adding another UNIX system or DOS partition, or enlarging an existing partition. Enlarging a partition requires reinstalling the operating system and (for a UNIX system partition) remaking the filesystem on the partition using `divvy(ADM)`.

### *DOS accessing utilities*

---

The DOS accessing utilities are discussed in detail in the `dos(C)` manual page. Note that you must have a bootable, although not active, DOS partition on the hard disk or a DOS floppy in order to use these UNIX system commands. For example, you can only transfer a file from a UNIX system partition on hard disk to a DOS floppy if either the DOS floppy is bootable or there is also a DOS partition on the hard disk.

You may also be able to use the UNIX system `dd(C)` and `diskcp(C)` commands to copy and compare DOS floppies. The UNIX system `dtype(C)` command tells you what type of floppies you have (various DOS and UNIX system types).

Also, the file `/etc/default/msdos` describes which DOS filesystems (for example, A:, B:, C:, ...) correspond to which UNIX system devices. The UNIX system does not record bad tracks in the DOS area of the hard disk. If a bad track develops in the DOS area, an operation such as `doscp` that attempts to access the affected area may fail. If such is the case, the message "Error on fixed disk" is displayed.

**NOTE** When trying to use the DOS utilities to access files on your DOS partition, you may see the error message "bad media byte". This message indicates that the DOS partition on the hard disk is not bootable. You can make your DOS partition bootable by first backing up the files on the DOS partition, booting DOS from the floppy, and formatting the DOS partition using the command:

```
format c: /s
```

You should now reinstall your DOS files.

### *File and directory arguments*

---

The file and directory arguments for DOS files take the form:

*device:name*

where *device* is a UNIX system pathname for the special device file containing the DOS diskette or DOS partition, and *name* is a pathname to a DOS file or directory. For example,

```
/dev/fd0:/john/memos
```

indicates that the file *memos* is in the directory */john*, and that both are in the device file */dev/fd0* (the UNIX system special device file for the primary floppy drive). Arguments without *device:* are assumed to be UNIX system files.

## User configurable default file

---

For convenience, the user configurable default file `/etc/default/msdos` can define DOS drive names that you can use in place of UNIX system special device file pathnames. For example, you can include the following entries in the above file:

```
A=/dev/fd096ds15
B=/dev/fd048ds9
C=/dev/dsk/0sC
D=/dev/dsk/0sD
```

Once you have defined the variables, you can use the drive letter A: in place of the special device file `/dev/fd0` (96ds15 by default) when referencing DOS files or directories. For example:

```
/dev/fd0:/john/memos
```

can be replaced with:

```
A:/john/memos
```

The drive letter B: refers to a low density (48ds9) primary floppy drive. Drive letter C: refers to the primary DOS partition on the primary hard drive. D: refers to a logical drive in the extended DOS partition.

**NOTE** If you get the message “cannot open `/dev/dsk/0sC`”, or a similar message, check the user permissions on the special device file involved. As super user, change the permissions with the `chmod` command. For example:

```
chmod 666 /dev/dsk/0sC
```

gives full read and write permissions to all users for the special device file `/dev/dsk/0sC`, which is the DOS partition on the primary hard disk.

## Mounting DOS filesystems on a UNIX system

---

In addition to the DOS utilities provided with the Operating System to manipulate DOS files, it is also possible to mount a DOS filesystem and access its files freely while still operating from your UNIX system.

This means that DOS files can be edited or examined in place, without first copying them into the UNIX filesystem. The major restriction is that DOS files and applications cannot be executed under this arrangement; this requires use of VP/ix (if running under your UNIX system) or booting of the DOS partition. However, data files and text files can be examined, copied or edited.

**NOTE** On a mounted DOS filesystem you cannot use the DOS utilities. If used, they return an error message:

```
dosdir: FAT not recognized on /dev/dsk/0sC
```

You cannot create DOS filesystems using the **mkfs(ADM)** command. The DOS mounting feature is intended for existing DOS filesystems (as in floppy disks and an existing DOS partition).

### *Configuring support for mounted DOS filesystems*

---

In order to mount DOS filesystems, the support for these features must be present in the kernel. If it is not, you must first add this to your kernel with the **mkdev(ADM)** command. Make certain you are logged in as *root* and enter the following command:

```
mkdev dos
```

△ **sysadmsh** users select: System ⇄ Configure ⇄ Kernel ⇄ DOS

This command adds the necessary functionality and prompts to relink the kernel. (If the link kit is not installed, you will be asked to install it.) After rebooting, you can mount DOS filesystems as described in the sections that follow.

### *How DOS filesystems are accessed*

---

The Operating System deals with DOS filesystems by superimposing certain qualities of UNIX system filesystems over the DOS filesystem without changing the actual files. UNIX system filesystems are highly structured and operate in a multiuser environment. Thus they include many distinctions that have no meaning under DOS, including:

- file ownership
- access permissions
- special files (pipes, device files, etc.)
- links

**NOTE** Other applications and operating systems permit the mounting and access of DOS filesystems in this manner. However, most modify the DOS filesystem in some way to accomplish this. To ensure portability, there are no proprietary modifications or extensions to the DOS filesystem in this implementation. The ability to mount these filesystems is achieved purely through the facilities of the filesystem switch (FSS).

In order to make DOS files readily accessible, access permissions and file ownership are superimposed on the DOS filesystem when mounted.

## Using the mount command

**NOTE** Make sure that you have run `mkdev dos` before attempting to use the `mount` command.

The form for a DOS filesystem mount command is:

```
mount -r -f DOS /dev/dsk/xsy /mountpoint
```

where:

*x* is the hard disk number  
*y* is the drive letter (C:, D:, etc.)  
*mountpoint* is the name of the directory in the root filesystem where the DOS filesystem is to be mounted

The `-r` flag mounts the filesystem read-only, an optional precaution that will prevent damage to the DOS filesystem, which is not as robust as a UNIX system filesystem.

**NOTE** DOS automatically calls the primary DOS drive, on the first disk, C:. If you have a primary DOS partition on the second disk this becomes D:, automatically, and logical drives on extended partitions are named in order, for example: disk0 Primary C: EXT E: F: G: H:, disk1 Primary D: EXT I: J:, etc. The naming convention in UNIX System V, for the above example, is as follows:

|      |     |     |     |     |     |     |     |     |
|------|-----|-----|-----|-----|-----|-----|-----|-----|
| DOS  | C:  | D:  | E:  | F:  | G:  | H:  | I:  | J:  |
| UNIX | 0sC | 1sC | 0sD | 0sE | 0sF | 0sG | 1sD | 1sE |

When using `mount`, you must give the specific hard disk and partition numbers (as opposed to using wildcards).

### Mounting a floppy disk

You can also mount DOS floppy disks, as in the following example using the 96-tpi floppy mounted on `/mnt`:

```
mount -r -f DOS /dev/fd096 /mnt
```

## Repairing and checking DOS filesystems

The operating system includes a DOS version of the `fsck(ADM)` utility that works on DOS filesystems. This utility reconciles the DOS FAT (File Allocation Table) to the files contained on the filesystem. When `fsck` is invoked, it automatically detects the DOS filesystem and invokes the proper binary.

## Who can access the mounted DOS filesystem

---

Only *root* can mount a filesystem. Access by users is governed by the permissions and ownership that *root* places on the DOS filesystem. Because of the limitations discussed earlier, DOS does not recognize permissions or ownership. When mounted on a UNIX system, the DOS files behave as follows:

- The permissions and ownership of the filesystem are governed by the mountpoint. For example, if *root* creates a mountpoint */x* with permissions of *777*, all users can read or write the contents of the filesystem. If the mountpoint is owned by *root*, all files within the DOS filesystem and any created by other users are all owned by *root*.
- The permissions for regular files will be either *0777* for readable/writable files or *0555* for read-only files. This preserves the consistency of the DOS filesystem. If a user can access the filesystem, the user will be limited by the permissions available under the DOS directory structure. This permission is read-only or read-write. When a file is created, the permissions are based on the *umask(C)* of the creator. For example, assume the user's *umask* is *022*, which generates files with permissions of *777*. Here are further examples.

### Example 1: Creating a file.

The permissions are based on the *umask* owner section. A *umask* of *022* will provide a file of *777* on the DOS partition. This is because the owner has not masked off the write bit for themselves.

### Example 2: Examining a file already on the DOS partition.

The permission you see is the logical AND of the UNIX system mountpoint permission and the DOS file permission. So, a UNIX system mountpoint of *750* and a DOS file permission of *555* will give you *550* for the permissions. This has nothing to do with the *umask*.

- There can only be one link for each file under the DOS filesystem. *."* and *.."* are a special case under this arrangement and are not links as they are on a UNIX system.
- On UNIX systems, features such as locking govern how, under certain programs and applications, a file is accessed simultaneously by different users. These features operate identically on a mounted DOS filesystem. Two users can edit the same file and write to it as permitted by the locking mechanism used.

## *Appearance of DOS files*

---

As no attempt is made to change the nature of DOS files, the carriage return character (^M) will be visible when editing a DOS file on a UNIX system. (UNIX systems use only a newline, while DOS uses a carriage return and a newline.) The `dtox(C)` and `xtod(C)` commands are the easiest way to switch the end-of-line format. `dtox` is used to change DOS format to UNIX system format, and `xtod` vice-versa.

## *Restrictions*

---

Additional logical restrictions that must be observed relating to filenames, modification times, and backup utilities are as follows.

### *File names*

The rules for file names and their conversion follows the guidelines found in the `dos(C)` manual page. In addition, the standard DOS restrictions on illegal characters apply. However, wildcards can be used just as they can with a UNIX system.

### *Modification times*

When accessed from the UNIX system partition, the creation, modification, and access times of DOS files are always identical and use GMT, or Greenwich Mean Time. (This is because UNIX System V uses GMT internally and converts it for the user.) This means that files created in the DOS filesystem while under a DOS or UNIX system will not have consistent times across the operating systems.

### *UNIX backup utilities*

The `backup(ADM)` utility cannot be used to make backups of a mounted DOS filesystem. DOS utilities and other copy programs like `tar(C)` will work as expected.

For more information, including more technical aspects of DOS usage, refer to `dos(C)`.



# *Administering DOS Services*

## Chapter 1

# Introduction

203

|                                  |     |
|----------------------------------|-----|
| Who should use this guide .....  | 203 |
| Organization of this guide ..... | 204 |

## Chapter 2

# Administering DOS Services

205

|                                                                     |     |
|---------------------------------------------------------------------|-----|
| Using the dosadmin program .....                                    | 206 |
| Adding and deleting user accounts .....                             | 206 |
| Administering the system console .....                              | 206 |
| Setting up the console .....                                        | 207 |
| DOS Services and display modes .....                                | 207 |
| Changing console display adapter cards .....                        | 209 |
| Using extended video modes .....                                    | 209 |
| Using the UNIX MultiScreen facility with DOS .....                  | 209 |
| Redefining the DOS Services switch-screen key sequence .....        | 209 |
| Administering terminals .....                                       | 210 |
| Setting up serial terminals .....                                   | 211 |
| Setting up PC scancode terminals .....                              | 211 |
| Administering COM ports .....                                       | 212 |
| Administering DOS printers .....                                    | 212 |
| Configuring the default DOS printer .....                           | 213 |
| Selecting a printer interface program .....                         | 213 |
| Changing the default DOS printer .....                              | 213 |
| Redirecting DOS printer output .....                                | 214 |
| Administering printers directly attached to DOS .....               | 214 |
| Changing the printer timeout .....                                  | 216 |
| Using the DOS MODE command .....                                    | 217 |
| Backing up filesystems .....                                        | 217 |
| Administering disk and diskette drives .....                        | 217 |
| Sharing diskette drives .....                                       | 218 |
| Diskette drives on very fast machines .....                         | 218 |
| Administering physical DOS partitions .....                         | 219 |
| Creating and formatting physical DOS partitions .....               | 220 |
| Default protection of DOS partitions .....                          | 220 |
| Changing the protection of DOS partitions .....                     | 222 |
| Administering virtual DOS partitions and virtual floppy disks ..... | 223 |

|                                                             |            |
|-------------------------------------------------------------|------------|
| Using dosadmin to create a virtual DOS partition .....      | 224        |
| Using virtual partitions .....                              | 225        |
| Administering virtual DOS partitions .....                  | 226        |
| Using dosadmin to create a virtual DOS floppy .....         | 226        |
| <b>Installing plug-in cards in your computer .....</b>      | <b>228</b> |
| Using directly attached DOS devices .....                   | 228        |
| Setting up /etc/dosdev .....                                | 231        |
| Mapping device names .....                                  | 232        |
| Limitations of direct attachment .....                      | 233        |
| <b>Making new DOS images .....</b>                          | <b>233</b> |
| Hardware changes that require new DOS images .....          | 234        |
| Standard ROMs and on-card ROMs .....                        | 234        |
| Changing special CONFIG.SYS commands .....                  | 235        |
| Adding device drivers to DOS images .....                   | 235        |
| Using mking to make new DOS images .....                    | 236        |
| Using dosadmin to make new DOS images .....                 | 236        |
| <b>System files affected by system administration .....</b> | <b>238</b> |

### *Chapter 3*

## ***Installing DOS applications***

**239**

---



---

|                                                                            |            |
|----------------------------------------------------------------------------|------------|
| <b>Tips on installing DOS applications to run under DOS Services .....</b> | <b>240</b> |
| Installing personal DOS applications .....                                 | 240        |
| Installing public DOS applications .....                                   | 242        |
| Defining the search path .....                                             | 243        |
| <b>Installing applications with advanced processor requirements .....</b>  | <b>243</b> |
| <b>Configuring DOS applications for use from the UNIX shell .....</b>      | <b>245</b> |
| Using dosadmin to configure DOS applications .....                         | 245        |
| Manually configuring DOS applications .....                                | 247        |
| <b>Installing copy-protected DOS applications .....</b>                    | <b>248</b> |
| Using a key disk .....                                                     | 249        |
| Using special installation procedures .....                                | 249        |
| Installing DOS applications on a DOS partition .....                       | 249        |
| Installing DOS applications under SCO systems .....                        | 249        |
| Installing applications under "raw" DOS .....                              | 250        |
| <b>Removing DOS applications .....</b>                                     | <b>252</b> |
| Removing the application from the fixed disk .....                         | 252        |
| Deleting UNIX links .....                                                  | 252        |
| Redefining the search path .....                                           | 252        |

## Chapter 4

# ***Tuning applications with the merge command*** **253**

---

|                                            |     |
|--------------------------------------------|-----|
| fastclk option .....                       | 254 |
| pollsleep option .....                     | 254 |
| merge network control options .....        | 255 |
| drive option .....                         | 255 |
| handle option .....                        | 256 |
| netbios option .....                       | 256 |
| 5f02 option .....                          | 257 |
| lpt_binary option .....                    | 257 |
| display option .....                       | 258 |
| return option .....                        | 258 |
| Using batch files for merge commands ..... | 258 |

## Chapter 5

# ***Tailoring the operation of DOS*** **261**

---

|                                                          |     |
|----------------------------------------------------------|-----|
| DOS defaults .....                                       | 262 |
| Using DOS options on the dos command line .....          | 262 |
| Installing DOS options to change defaults .....          | 263 |
| Installing DOS options to override system defaults ..... | 264 |
| Using dosadmin to install and remove DOS options .....   | 265 |
| Characteristics you can tailor using dosadmin .....      | 265 |
| Startup memory .....                                     | 265 |
| DOS startup file .....                                   | 265 |
| DOS device file .....                                    | 266 |
| Using dosadmin to tailor DOS .....                       | 266 |
| Selecting the configuration menu .....                   | 267 |
| Selecting the DOS command or application .....           | 268 |
| Filling in the configuration menu .....                  | 269 |
| Using dosopt to install and remove DOS options .....     | 271 |
| Creating a home directory options file .....             | 273 |
| Installing DOS options .....                             | 273 |
| Displaying installed options .....                       | 274 |
| Removing installed options .....                         | 275 |
| Restoring DOS options files .....                        | 275 |
| Allocating memory for DOS .....                          | 276 |
| Running RAM-hungry applications in base memory .....     | 276 |

|                                       |     |
|---------------------------------------|-----|
| Preparing minimal startup files ..... | 277 |
| Using minimal startup files .....     | 277 |
| Allocating memory for windows .....   | 278 |
| Memory for Windows 3.0 .....          | 278 |
| Memory for Windows 3.1 .....          | 279 |

## Chapter 6

# DOS options

281

---



---

|                                                            |     |
|------------------------------------------------------------|-----|
| Attach devices to DOS .....                                | 284 |
| Default devices .....                                      | 284 |
| The keyboard .....                                         | 284 |
| The fixed disk .....                                       | 284 |
| Diskette drives .....                                      | 284 |
| The mouse .....                                            | 285 |
| Physical DOS partitions .....                              | 285 |
| Display adapters .....                                     | 285 |
| Other standard DOS devices .....                           | 286 |
| COM ports .....                                            | 286 |
| Expanded Memory (EMS) .....                                | 287 |
| Printer ports .....                                        | 287 |
| The game port .....                                        | 287 |
| Attaching DOS drives .....                                 | 288 |
| Specifying immediate, exclusive, or read-only access ..... | 289 |
| Specifying custom devices .....                            | 289 |
| Canceling device specifications .....                      | 290 |
| Using -a to prevent DOS from accessing devices .....       | 290 |
| Using -a to remove defaults from options files .....       | 291 |
| Display-oriented and stream-oriented DOS programs .....    | 291 |
| Pass command directly to DOS command interpreter .....     | 293 |
| Set initial current drive .....                            | 294 |
| Interpret configuration file .....                         | 294 |
| Help text .....                                            | 295 |
| Alternative DOS load image .....                           | 296 |
| DOS memory size .....                                      | 297 |
| Run autoexec.bat .....                                     | 298 |
| Change DOS printer timeout .....                           | 299 |
| Translate DOS switch characters and path separators .....  | 300 |
| Dosopt "verbose" mode .....                                | 301 |
| Set X Client startup .....                                 | 302 |

|                                      |            |
|--------------------------------------|------------|
| <b>dosopt null option .....</b>      | <b>302</b> |
| <b>Remove assigned options .....</b> | <b>302</b> |

## Chapter 7

# **Using DOS Services NLS features 305**

|                                                                    |            |
|--------------------------------------------------------------------|------------|
| <b>Features .....</b>                                              | <b>305</b> |
| <b>Prerequisites .....</b>                                         | <b>306</b> |
| <b>Theory of National Language Support .....</b>                   | <b>307</b> |
| <b>DOS code pages and UNIX operating system code sets .....</b>    | <b>308</b> |
| <b>Differences between DOS code pages and UNIX code sets .....</b> | <b>308</b> |
| <b>How MS-DOS handles NLS .....</b>                                | <b>309</b> |
| <b>Varieties of DOS code pages .....</b>                           | <b>309</b> |
| <b>Code page switching .....</b>                                   | <b>310</b> |
| <b>Tailoring DOS for different languages .....</b>                 | <b>310</b> |
| <b>How the UNIX operating system handles NLS .....</b>             | <b>310</b> |
| <b>How DOS Services handles NLS .....</b>                          | <b>311</b> |
| <b>Display considerations .....</b>                                | <b>311</b> |
| <b>Setting up the DOS Services NLS environment .....</b>           | <b>312</b> |
| <b>Setting UNIX operating system environment variables .....</b>   | <b>314</b> |
| <b>DOS National Language Support codes .....</b>                   | <b>314</b> |
| <b>NLS commands supported by DOS Services .....</b>                | <b>315</b> |
| <b>DOS Services exceptions .....</b>                               | <b>316</b> |
| <b>Sample NLS setup procedures for VGA displays .....</b>          | <b>316</b> |
| <b>Setting up the startup files .....</b>                          | <b>316</b> |
| <b>Using DOS images .....</b>                                      | <b>317</b> |
| <b>Resetting the DOS image .....</b>                               | <b>318</b> |
| <b>Sample NLS setup procedures for ISO terminals .....</b>         | <b>319</b> |
| <b>Naming DOS and UNIX operating system files .....</b>            | <b>320</b> |
| <b>Printing in an NLS environment .....</b>                        | <b>321</b> |
| <b>Converting text files .....</b>                                 | <b>322</b> |
| <b>Using dos2unix and unix2dos .....</b>                           | <b>322</b> |
| <b>Using charconv .....</b>                                        | <b>322</b> |
| <b>charconv options .....</b>                                      | <b>323</b> |
| <b>Using the CONVOPTS environment variable .....</b>               | <b>324</b> |

---

|                                                                    |     |
|--------------------------------------------------------------------|-----|
| <b>Changing colors in DOS windows</b> .....                        | 326 |
| How DOS and X Windows handle colors .....                          | 326 |
| Changing the color scheme of your windows .....                    | 327 |
| Installing the DOS colors .....                                    | 328 |
| <b>X System resources for DOS Services</b> .....                   | 328 |
| Running DOS Services on remote servers .....                       | 330 |
| Setting the XMERGE environment variable .....                      | 330 |
| DOS Services fonts .....                                           | 330 |
| <b>Using expanded memory (EMS)</b> .....                           | 331 |
| <b>Using multiport serial cards</b> .....                          | 331 |
| <b>DOS Services/DOS compatibility</b> .....                        | 332 |
| DOS interrupts .....                                               | 332 |
| DOS clock interrupt rate .....                                     | 333 |
| Applications that continuously poll the keyboard .....             | 333 |
| Configuring applications that buffer keystrokes .....              | 333 |
| Local and remote drives and file handles .....                     | 333 |
| Limits on number of open files .....                               | 334 |
| Protected mode restrictions .....                                  | 334 |
| <b>Passing UNIX environment variables to DOS</b> .....             | 335 |
| Setting your DOS search path with DOSPATH .....                    | 336 |
| Setting other DOS environment variables with DOENV .....           | 337 |
| <b>Running DOS or a DOS application as the default shell</b> ..... | 339 |
| Running DOS as the default shell .....                             | 339 |
| Running a DOS application as the default shell .....               | 339 |
| <b>Scheduling DOS programs</b> .....                               | 340 |

---

|                                                                     |     |
|---------------------------------------------------------------------|-----|
| <b>Characteristics of DOS on the console</b> .....                  | 341 |
| <b>Using EGA cards</b> .....                                        | 342 |
| Prerequisites to using EGA cards .....                              | 342 |
| Using EGA "fast" and "reliable" modes .....                         | 343 |
| DOS processes that suspend when their screens are not visible ..... | 344 |
| <b>Using VGA cards</b> .....                                        | 344 |
| <b>Using PC scancode terminals</b> .....                            | 344 |

|                                          |     |
|------------------------------------------|-----|
| Using serial terminals .....             | 345 |
| Serial terminal keyboard mapping .....   | 345 |
| Console and terminal control codes ..... | 349 |

## Appendix B

# Terminfo guide

351

|                                                        |     |
|--------------------------------------------------------|-----|
| <b>Basic concepts</b> .....                            | 352 |
| terminfo .....                                         | 352 |
| terminfo files .....                                   | 352 |
| Standard and PC scancode terminals .....               | 353 |
| Escape sequences .....                                 | 354 |
| The DOS Services lead-in sequence .....                | 354 |
| Escape sequences used by DOS Services .....            | 354 |
| <b>Output on standard terminals</b> .....              | 355 |
| Special terminfo descriptors .....                     | 358 |
| pctrm .....                                            | 358 |
| dispc .....                                            | 359 |
| How PC characters are displayed on terminals .....     | 360 |
| How PC characters are displayed on ISO terminals ..... | 360 |
| Combined ISO and PC character set terminals .....      | 361 |
| <b>Input on standard terminals</b> .....               | 362 |
| Default "key emulation" escape sequences .....         | 363 |
| Default "special action" escape sequences .....        | 364 |
| Using terminfo descriptors for key emulation .....     | 364 |
| pctrm .....                                            | 365 |
| Changing the lead-in character .....                   | 366 |
| <b>About PC scancode terminals</b> .....               | 366 |
| Switching commands .....                               | 367 |
| Alternate flow control .....                           | 367 |
| Serial line parameters .....                           | 367 |
| pctrm .....                                            | 368 |
| Sequence of use .....                                  | 369 |
| SCO scancode terminal support .....                    | 370 |
| <b>Output on PC scancode terminals</b> .....           | 370 |
| Parameterized strings .....                            | 371 |
| dispc as a parameterized string .....                  | 371 |
| Code pages .....                                       | 372 |
| <b>Input on PC scancode terminals</b> .....            | 373 |
| Changing the lead-in sequence .....                    | 373 |



|                                                |            |
|------------------------------------------------|------------|
| <b>PC-character-to-ASCII conversions .....</b> | <b>375</b> |
| <b>terminfo escape sequences .....</b>         | <b>376</b> |
| <b><i>Index</i> .....</b>                      | <b>997</b> |

## Chapter 1

# Introduction

---

This guide covers administrative and advanced topics of DOS Services. Administering DOS Services is no different in most respects from administering separate, conventional DOS and UNIX systems. The administrator's responsibilities include installing and maintaining system hardware and software, regularly backing up system data, assisting users, and informing them of changes to the system.

This guide supplements the system administration instructions in your DOS and UNIX documentation. You should be familiar with that documentation because this guide is not a comprehensive description of the system administrator's responsibilities. In general, you administer DOS Services by using UNIX procedures to accomplish UNIX tasks and DOS procedures to accomplish DOS tasks.

## *Who should use this guide*

---

This guide is for the SCO Open Desktop/SCO Open Server system administrator, the person responsible for maintaining the day-to-day operation of the system. This guide covers the administrative procedures that are necessary to manage the combined DOS and UNIX environment of the SCO system. It supplements the documentation on your computer hardware, DOS, and the UNIX System.

This guide is also intended for advanced DOS Services users who would like to know more about customizing the DOS environment and maximizing its efficiency.

## *Organization of this guide*

---

This guide has seven additional chapters and two appendixes:

Chapter 2: "Administering DOS Services," tells you how to manage user accounts, set up and administer computer hardware used by DOS, and configure your computer's resources to meet the combined needs of DOS and UNIX System users.

Chapter 3: "Installing DOS applications," provides hints for installing DOS applications for personal or public use, installing copy-protected applications, setting up DOS applications for use from the UNIX shell, and removing DOS applications.

Chapter 4: "Tuning Applications with the `merge` Command," tells you how to tune the operation of the DOS process for certain applications.

Chapter 5: "Tailoring the Operation of DOS," and Chapter 6: "DOS Options," tell you how to modify DOS characteristics and configure DOS to suit your needs.

Chapter 7: "Using DOS Services in a Multilanguage Environment," describes DOS Services features that support multiple languages in an environment shared by DOS and the UNIX operating system..

Chapter 8: "Other Advanced Topics," covers a wide variety of techniques for customizing DOS and improving DOS Services efficiency.

Appendix A: "Using the System Console and Terminals," explains DOS Services's operation with different console display adapters and terminals.

Appendix B: "terminfo Guide," explains the function of the `terminfo` file and tells you how to use it to run DOS Services on the terminals attached to your UNIX system.

## Chapter 2

# Administering DOS Services

---

This chapter covers the topics that are essential for using your computer as a combined DOS and UNIX machine. The topics include:

- using the `dosadmin` program,
- adding and deleting user accounts,
- configuring applications that buffer keystrokes,
- administering the system console,
- administering terminals,
- administering COM ports,
- administering DOS printers,
- backing up filesystems,
- administering disk and diskette drives,
- administering physical DOS partitions,
- administering virtual DOS partitions and floppy disks,
- installing plug-in cards,
- making new DOS images, and
- system files affected by system administration.

To administer DOS Services effectively, you should be familiar with the contents of the DOS sections in the User's Guide, in addition to this chapter.

Most of the descriptions in this chapter assume that you are logged in as root or you are the super user. The UNIX `#` prompt is therefore shown in most examples. Examples that apply to any user are shown with appropriate DOS or UNIX (`$`) prompts.

### *Using the dosadmin program*

---

The **dosadmin** menu system provides an easy way to change the values of three important DOS characteristics: memory, DOS startup files, and DOS device files. Throughout this guide you will use the **dosadmin** menu to perform various system administration tasks.

You can run **dosadmin** from either the DOS environment or the UNIX shell. When you start **dosadmin** while using the DOS environment, your prompt before starting **dosadmin** and after exiting **dosadmin** is a DOS prompt, such as C>. When you start **dosadmin** from the UNIX shell, your prompt before and after using **dosadmin** is \$ (or # if you are logged in as root). The actual **dosadmin** procedures and displays are identical, whether you start **dosadmin** from the UNIX shell or in the DOS environment.

### *Adding and deleting user accounts*

---

DOS Services requires no special procedures for adding or deleting user accounts. Any user with a valid UNIX account can use DOS Services. You are not required to be logged in as root or have any special permissions, for example, to run DOS or use DOS and UNIX commands.

In general, follow the instructions in the Administrator's Guide for adding, deleting, and administering user accounts.

Although no special configuration files or setup procedures are required to use DOS, SCO Open Desktop/SCO Open Server users may want to customize the way DOS runs. Users who follow the instructions in the DOS sections can alter the behavior of DOS in many ways without help from the system administrator. If users at your site are unfamiliar with the DOS or UNIX systems, they may require your assistance. The User's Guide contains hints for system administrators who want to modify DOS Services defaults or help users configure their own individual environments.

### *Administering the system console*

---

DOS Services works with any system console that consists of a monochrome or color monitor with a PC-compatible keyboard, connected to a monochrome, Hercules®, CGA, or VGA display adapter.

- Setting Up the Console
- Changing Console Display Adapter Cards

- Using Extended Video Modes
- Using the UNIX MultiScreen™ Facility with DOS
- Redefining the DOS Services Switch-Screen Key Sequence

## *Setting up the console*

---

No special procedures are required to set up the system console for use with DOS Services. The only requirement for using DOS Services is that you must use one of the display adapters that DOS Services recognizes. These include monochrome, Hercules, CGA, and VGA adapters.

## *DOS Services and display modes*

---

When you use DOS Services, you can run DOS applications on serial terminals or on a console that uses a monochrome, Hercules, CGA, or VGA display adapter and a compatible display. In addition, by using different DOS images, DOS Services can simulate the characteristics of display adapters less powerful than your actual physical display adapter. (Refer to “Making new DOS images” later in this chapter for more information on DOS images.)

This flexibility can lead to unexpected problems if you are not careful how you install, configure, and run graphics applications. For example, an application may fail to run if you have configured it to use a different type of display than you really use. If your system uses more than one type of display, you may find that some applications run correctly on some displays but not on others.

Some applications automatically detect the type of display adapter being used and adjust themselves to display graphics data correctly. Other applications do not work properly unless you identify your display adapter when you install them. When you install these applications, be sure you follow the application manufacturer’s instructions and configure the applications to work with the display adapter you intend to use.

If you run DOS processes from a serial terminal, identify your display as a monochrome display adapter (MDA).

If you use a graphics display adapter, you should normally configure your DOS applications to use the type of adapter in your computer, provided the application is compatible with that type of adapter. For example, if you use a VGA card, configure your DOS applications to use a VGA card.

In rare cases, you may run into difficulty because you are trying to use an incompatible combination of physical display adapter, DOS image, and application. By default, DOS Services uses DOS images that correspond to your

physical display — VGA images for VGA displays, CGA images for CGA displays, and so on. Most DOS applications that are configured for less powerful display adapters run correctly, automatically, when you run them on more powerful adapters. For example, an application configured for CGA displays may run correctly on a VGA display. However, some applications fail when run on a more powerful display than they are configured for. If you run into this restriction, you can work around it in either of two ways:

- Reconfigure the application so it expects to be run on the display adapter you are using. For example, if your application expects to be run on a CGA card and you use a VGA card, reconfigure the application to expect a VGA card.
- Use a DOS image that corresponds to the display adapter the application expects to use.

For example, if you have a VGA card but your application expects to use a CGA card, you can start a DOS environment with the command:

```
dos +acga
```

Then start your application. If you want to start the application directly from the UNIX shell, you can also use a command such as:

```
dos +acga appl
```

where **appl** is the command that starts your application. These commands cause DOS Services to use a CGA image instead of the default image, and your application views your computer hardware as though you actually have a CGA card.

Note that this technique works only if your physical display is capable of using the DOS image you request. The following table shows usable combinations of physical display adapters and DOS images. Fields with a dash ( - ) indicate unusable combinations.

| Option | Capability By Physical Display Type |          |      |      |
|--------|-------------------------------------|----------|------|------|
|        | Monochrome                          | Hercules | CGA  | VGA  |
| None   | mono                                | herc     | CGA  | VGA  |
| +amono | mono                                | mono     | mono | mono |
| +aherc | -                                   | herc     | -    | -    |
| +acga  | -                                   | -        | CGA  | CGA  |
| +avga  | -                                   | -        | -    | VGA  |

Refer to Chapter 6 for more information on the **+a** option. Refer to Appendix A for more information on the characteristics of graphics displays.

## *Changing console display adapter cards*

---

If you change your console display adapter card after you install DOS Services, you may need to make a new DOS image. See "Making New DOS Images," later in this chapter, for further information.

## *Using extended video modes*

---

Whether or not you can use extended video modes on VGA cards depends on how your DOS images were created. If you have a VGA display and want to run high-resolution graphics applications, you need to create a VGA DOS image. You can run DOS applications that require monochrome or CGA displays using your VGA card even if you do not make a VGA image.

To make a VGA image, your computer must have a 100 percent IBM-compatible VGA card. If your card is not fully IBM-compatible, the procedure for making a VGA image may fail. When the procedure fails, symptoms range from an error message like the one shown below to a system lockup.

```
VGA Image was not made.
(Incompatible VGA card?)
(1)
```

For information on how to create a new DOS image, refer to "Making New DOS Images," later in this chapter.

## *Using the UNIX MultiScreen facility with DOS*

---

DOS Services is compatible with the standard UNIX MultiScreen facility. If your console can display multiple, independent screens, you can run DOS commands and applications or a DOS environment in any screen.

When you run multiple DOS processes in separate screens, the default key sequence for switching between screens is `<Alt><Ctrl><Fn>`, where `<Fn>` is a function key (such as `<F1>` or `<F2>`). `<Alt><Ctrl><F2>` switches to screen 2, `<Alt><Ctrl><F3>` switches to screen 3, and so on. The UNIX system also uses the `<Alt><Ctrl><Fn>` convention for switching between screens, so you can switch between any active DOS and UNIX screens using `<Alt><Ctrl><Fn>`.

## *Redefining the DOS Services switch-screen key sequence*

---

You can redefine the switch-screen key sequence that applies to DOS screens if `<Alt><Ctrl><Fn>` does not suit your needs. For example, you may have a DOS application that does not work properly with the default switch-screen sequence because it assigns `<Alt><Ctrl><Fn>` a special meaning. You can redefine the switch-screen key sequence to use a function key together with any combination of the `<Ctrl>`, `<Alt>`, and `<Shift>` keys, or to use only a function key. To



redefine the switch-screen key sequence, use the **switchkey** command with the syntax:

**switchkey [-cas]**

In this syntax, **c** stands for the (Ctrl) key, **a** stands for the (Alt) key, and **s** stands for the (Shift) key. You can use these options in any combination and specify them in any order. To switch screens any time after you use the **switchkey** command, use the keys you specified with **switchkey** in addition to a function key. For example, to specify that you want to require (Ctrl) and (Shift) along with a function key, type:

**\$ switchkey -cs**

Thereafter, to switch screens, press (Ctrl)(Space)(Fn)

To specify that you want to use only function keys, without (Ctrl), (Alt), or (Shift), use **switchkey** with only a hyphen as an argument:

**\$ switchkey -**

Thereafter you can use (F1), (F2), (F3), and so on, without additional keys, to switch from your current DOS screen to a different one. The **switchkey** command with no arguments displays the current switch-screen key sequence.

The switch-screen key sequence you define with **switchkey** applies only when you are viewing a DOS screen. When you are viewing a UNIX screen, the existing UNIX switch-screen sequence is effective.

DOS Services and the X Window System both use the same default switch-screen key sequence and are affected the same way when you use **switchkey**.

Any switch-screen key sequence that you define with **switchkey** applies to any DOS or X Window sessions run on the console where you ran **switchkey**. DOS and X Window sessions that are currently running when you redefine the switch-screen key sequence are also affected. Your specified key sequence remains effective until you issue another **switchkey** command or reboot your computer. To make sure your preferred key sequence remains effective even after a reboot, include the **switchkey** command in */etc/profile* or your home directory *.profile*.

## Administering terminals

---

DOS Services works with a variety of terminals. Appendix A of this guide discusses using terminals. This section tells how to set up serial and PC scan-code terminals. See the *Release Notes* for a list of supported terminals. Refer to "Appendix B" in this manual for more information on the UNIX *terminfo* file and how DOS Services uses it to run DOS on terminals attached to the system..

## Setting up serial terminals

---

No special procedures are required to set up serial terminals for use with DOS Services. Simply follow the terminal and UNIX system manufacturers' instructions for setting up the terminal for standard UNIX use. The procedures require physically attaching the terminal to a serial port in your system unit and making sure the UNIX **TERM** environment variable is set correctly for anyone who uses the terminal.

You may also need to modify entries in your *terminfo* file. See Appendix B for a full description.

## Setting up PC scancode terminals

---

DOS Services supports PC scancode terminals that can be switched between character and scancode modes under software control. These terminals are referred to as "switchable scancode terminals." Wyse® Technology's WY-60 and Kimtron® Corporation's Model KT-70/PC terminals are examples of switchable scancode terminals. DOS Services automatically switches scancode terminals into character mode when they communicate with the UNIX system and into scancode mode when they communicate with DOS.

To set up a scancode terminal, follow these procedures:

1. Configure your scancode terminal to use eight data bits and one stop bit, no parity, and 25-line display mode.
2. Configure the terminal to use normal character (ASCII) mode by default. (DOS Services switches these terminals to scancode mode automatically when necessary.) Follow the steps that apply to your terminal:
  - Wyse WY-60 terminals: Set the terminal's "Personality" to WY-60.
  - Kimtron KT-70/PC terminals: Set the "Emulation" parameter to "IX" mode.
3. Attach your terminal to any available serial line on your computer. Make sure the serial line is configured for eight data bits, one stop bit, and no parity.
4. Define the **TERM** environment variable appropriately for the terminal you are using:
  - Use the name **wy60ix** for Wyse Technology's WY-60 terminal.
  - Use the name **kt70pcix** for Kimtron Corporation's Model KT-70/PC.

Refer to the manual for your terminal for more information on these procedures.

## *Administering COM ports*

---

On the SCO system, both the UNIX environment and DOS can use COM (serial) ports, but only one process (either UNIX or DOS) can access a particular COM port at one time. If you plan to use a COM port for DOS work, follow these procedures:

1. Make sure no `getty` is running on the physical device (for example, `/dev/tty1a` or `/dev/tty2a`) that you want to make available to DOS as a COM port. To do this, log in as root or become the super user and run the `disable` command. For example, to disable the `getty` process on `/dev/tty1a`, type:

```
disable /dev/tty1a
```

2. Use the `chmod` command to set permissions of the device to be readable and writable. If you want to use `/dev/tty1a`, you would type:

```
chmod 666 /dev/tty1a
```

The COM port is now ready for use with DOS. You can attach it to any DOS session with the `+a` option:

```
dos +acom1
```

To return the COM port for use by the UNIX system, type:

```
enable /dev/tty1a
```

For more information on using COM ports with DOS, refer to Chapter 6 in this guide and to the DOS sections in the User's Guide.<sup>1</sup>

## *Administering DOS printers*

---

The DOS section of the User's Guide describes, from the user's point of view, how to print using DOS commands or applications. This section explains how to set up and administer DOS printers. It covers the following topics:

- configuring the default DOS printer,
- changing the default DOS printer,
- redirecting DOS printer output, and
- administering printers directly attached to DOS.

---

1. "Smart" cards are not supported by DOS Services unless connected through a special driver which is provided by the manufacturer. See "COM Ports" in Chapter 6.

## Configuring the default DOS printer

---

By default, DOS Services sends all DOS printing via the UNIX spooler to a printer named **doslp**. The DOS Services installation routine automatically attempts to configure your default UNIX printer, if one exists, so it can also be used as the default DOS printer, **doslp**. To determine whether **doslp** has been configured, type the following **lpstat** command:

```
lpstat -p doslp
```

This command tells you whether or not **doslp** is properly configured. If it is not, you need to configure **doslp** yourself.

To configure **doslp**, use the UNIX **sysadmsh** command. Refer to the Administrator's Guide for complete instructions on configuring printers. You need to know the UNIX device name of the printer you want to use as **doslp** (for example **/dev/lp0**). Use standard UNIX procedures to perform the following operations:

1. Assign the printer name **doslp** to your preferred printing device.
2. Choose the appropriate printer interface program as described below. (The interface program is also known as a printer model.)

### Selecting a printer interface program

DOS Services supplies a printer interface program named **dosmodel**. It is the same as the **standard** model, except that it does not print banner pages at the beginning of each print job, and it does not output a form feed at the end of each print job.

If DOS Services automatically configured **doslp** during the installation procedure, it used **dosmodel**. Any printer that recognizes the **standard** model can also use **dosmodel** for printing DOS output or any other output that does not require banner pages or form feeds.

You can choose the **standard** model for **doslp** if you prefer to have banner pages and automatic form feeds.

If your printer does not recognize the **standard** model, you should not use either the **standard** model or **dosmodel**. Instead, choose a model that is appropriate for your printer.

## Changing the default DOS printer

---

The default DOS printer is always named **doslp**. To change the physical printer that **doslp** refers to, follow the standard UNIX procedures for configuring printers. Assign the name **doslp** to your preferred printing device and choose an appropriate interface program.

### *Redirecting DOS printer output*

---

You can use printers other than **doslp** for DOS printing. You can select any convenient printer name and any appropriate interface program when you configure a printer.

To use any available UNIX printer for DOS printing, use the DOS Services **printer** command to correlate a print stream with a particular UNIX printer and print command. The syntax for selecting a print stream, a printer, and a UNIX print command is:

**printer** [*print\_stream*] *unixprint\_command*

where *print\_stream* is LPT1, LPT2, or LPT3 and *print\_command* is a UNIX command that processes the specified print stream. If you do not specify a print stream, **printer** assumes LPT1 by default. Use the **printer** command in the DOS environment. Assume, for example, that you want to direct DOS printer output sent to LPT2 to a UNIX printer named "laser." Follow these steps:

1. Start a DOS environment if you have not already started one.
2. Use the following **printer** command to direct print stream LPT2 to the printer named "laser":

**C: printer lpt2 unix "lp -dlaser"**

In this command, the **-d** ("destination") option to the **lp** command identifies the printer named "laser."

3. To send DOS printer output to the printer, name the DOS print stream in your DOS print command. For example:

**C: copy letter.txt lpt2**

You can direct printer output from DOS applications to any UNIX printer using the same procedures.

If you want a **printer** command to be effective every time you use the DOS environment, you can include it in an **AUTOEXEC.BAT** file.

Note that the *print\_command* that you specify when you use the **printer** command is typically **lp -dprinter**, where *printer* is the name of a UNIX printer. However, *print\_command* can be any UNIX command that you choose to use to process a print stream.

### *Administering printers directly attached to DOS*

---

Under some circumstances, SCO system users may not want to use the UNIX spooling system when they use DOS printing. In these cases, users can attach a printer directly to a DOS process. The printer is then under the direct control of DOS, and not available for UNIX printing.

If a user wants to attach a printer directly that is currently configured for UNIX printing, you need to disable UNIX printing on that printer. Follow these procedures:

1. Log in as root or become the super user.
2. Disable UNIX printing by using the **disable** command. For example, if you want to disable UNIX printing on the printer named **doslp**, issue the command:

```
disable doslp
```

Note that while UNIX printing is disabled, users can continue to spool print jobs to that printer. However, the jobs are not printed until the printer is re-enabled for UNIX use.

3. To directly attach a printer to a DOS process, start DOS using the **+a** option to specify the physical printing device you want to use. DOS Services uses the device names **lp0**, **lp1**, and **lp2** to identify the first, second, and third parallel printer ports. Use the name that corresponds to the port your printer is attached to. For example, if your printer is attached to **/dev/lp0**, you can start the DOS environment with the command:

```
$ dos +alp0
```

Consult the manuals for your computer and the DOS Services **/etc/dosdev** file if you need assistance in identifying your printer port.

4. The printer can now be used with any of the standard DOS printing techniques, including the **PRINT** command, copying a file to the printing device, and printing from an application.

The DOS name of the printing device is determined by DOS, which recognizes a maximum of three printing devices. These are always named **LPT1**, **LPT2**, and **LPT3**. The UNIX system has fewer restrictions — in theory, any number of printer ports can exist, and they can be named in any fashion. Nevertheless, DOS Services recognizes only three parallel printer ports and names them **lp0**, **lp1**, and **lp2**.

Notice that the DOS name does not necessarily correspond to the name of the physical port attached in the previous step.

DOS checks each of the directly attached physical printer ports, in order, to identify the first port connected to a printer. No matter which physical port the printer is attached to, DOS names the first printer it finds "LPT1," the second printer it finds "LPT2," and so on. If you have only one printer and it is attached to the physical port named **lp2**, DOS views this printer as **LPT1**.

For example, if you have one directly attached printer, you can print a file with the command:

```
C: copy filename lpt1
```

When your DOS session ends, the printer can be enabled again for UNIX printing.

**NOTE** When you directly attach a printer to a DOS process as described in this section, you can use all DOS PRINT command options when you use that printer.

5. When the user finishes the printing that required directly attaching the printer to DOS, you can use the **enable** command to re-enable UNIX printing. For example, to re-enable the printer **doslp**, issue the command:

**# enable doslp**

**NOTE** A single physical printer may have more than one printer name associated with it. If the UNIX printer you are disabling has more than one UNIX printer name, then, in step 2, you should issue the **disable printer\_name** command for each printer name correlated with that printer. Similarly, in step 3 of this procedure, you may need to issue multiple **enable** commands.

### *Changing the printer timeout*

---

When you use a DOS application that prints, DOS Services by default sends DOS printer output to the UNIX spooler when either of the following conditions is true:

- You exit the application and return to your DOS or UNIX prompt.
- More than 15 seconds have elapsed since the application has sent a character to be printed.

This default is convenient for most applications. However, under certain conditions you may want to change the printer timeout from 15 seconds to a different value. For example, if your application pauses for more than 15 seconds while printing, the printer output sent before and after each pause is spooled as a separate UNIX print job.

To change the printer timeout in the DOS environment, use the DOS Services **printer** command with the syntax:

**printer [lptn] unix /t[timeout]**

In this command, **LPTn** identifies the print stream for which you wish to change the timeout. **LPT1** is assumed if you do not specify a print stream. **timeout** is a value, in seconds, between 5 and 3600. For example, if you want to increase the timeout for print stream **LPT2** to one minute, type the command:

**C: printer lpt2 unix /t60**

Using the **/t** option without specifying a value returns the printer timeout for the specified print stream to the default of 15 seconds.

The timeout value "0" has a special meaning: it delays the spooling of DOS printer output until you exit the application.

You can use the `ts` option to the `dos` command instead of the `PRINTER` command to control the printer timeout. Refer to Chapter 6 for information on the `ts` option.

## *Using the DOS MODE command*

---

If you use the `MODE` command to configure a DOS printer, the command has no effect unless the printer is directly attached to the DOS process.

## *Backing up filesystems*

---

To guard against permanently losing important data, you should regularly back up all data on your fixed disk. DOS Services requires no special procedures for system backups. Simply use UNIX backup procedures for the filesystem shared by DOS and the UNIX operating system. and DOS backup procedures for the physical DOS partition(s).

Consult your DOS and UNIX documentation for specific procedures recommended for your system.

**NOTE** If they are stored on a physical or virtual DOS partition, some copy-protected DOS applications cannot be backed up and restored using the `BACKUP` and `RESTORE` commands. For more information, consult your DOS user's manual and the instructions for your applications.

## *Administering disk and diskette drives*

---

Your DOS Services system may have one or more diskette drives that can be assigned to either the DOS or the UNIX environment. A diskette drive mounted as a UNIX device is assigned to the UNIX environment and is not accessible as a DOS diskette drive. If you insert a diskette containing a UNIX filesystem into a diskette drive and mount it as a UNIX device, any files on that diskette become part of the filesystem shared by DOS and the UNIX operating system.. You can then use DOS drive C: to access files on the mounted diskette in the same way you access files in the fixed disk filesystem shared by DOS and the UNIX operating system..

To be directly accessible to DOS as DOS drives A: or B:, diskette drives must not be mounted as UNIX devices or in use by UNIX programs (such as `cpio`). If you have more than one diskette drive on your system, you must not have any of them mounted as UNIX devices or otherwise in use by the UNIX system



when you attempt to access a diskette drive as DOS drive A: or B:. DOS Services prevents a DOS process from using all diskette drives as long as any other process is using any diskette drive.

### *Sharing diskette drives*

---

When diskette drives are not mounted as UNIX devices or otherwise used by the UNIX system, DOS users can access them on a first-come, first-served basis. Whenever a diskette drive has not been used for five seconds or more, it is available to the first DOS process that accesses that drive.

### *Diskette drives on very fast machines*

---

In a standard DOS machine, each ROM Bios chip is specially designed to accommodate the hardware used by that manufacturer. To make DOS run under UNIX, DOS Services emulates these ROM Bios routines, but must run on whatever hardware is being used by the host system.

A few of the newest machines are so fast that DOS Services may not read the diskettes accurately. On machines faster than 66 Mhz, or those with processors newer than the 80486, users may see one of the following messages when trying to read that drive.

```
Not ready reading drive A:
Abort, Retry, Fail?
```

```
I/O error reading drive A
```

The system administrator may find evidence in system logs that some process is timing out repeatedly while waiting for the drive to respond. This behavior is usually consistent rather than intermittent.

DOS Services offers the **patchflop** utility to correct this problem if it occurs. It "patches" the operation of the routines that read the floppy drive. The syntax for **patchflop** is as follows:

```
./patchflop [-p] [value]
```

The **-p** option prints the current setting used to adjust the ROM BIOS timing loops. The default value is 100.

The *value* argument is a single decimal number that sets the timing loops in the ROM BIOS. The number you need to use corresponds roughly to the MHz rating for your machine.

To use **patchflop**, assume root privileges and move to the directory where both the *bios.rom* file and the **patchflop** utility are stored.

```
cd /usr/lib/merge
```

The repair procedure is a trial and error process. The **patchflop** value does not take effect until a new DOS session is started.

First use the **-p** option to display the current setting:

```
./patchflop -p
```

For your first attempt, start with the MHz rating of your machine. For example:

```
./patchflop 50
```

Every time you use **patchflop** to modify the timing loops in the *BIOS* ROM file, start a new DOS session, and see if the floppy(s) can now be reliably accessed. If not, exit the DOS session, and run **patchflop** with a larger number. Increment the number by units of 5 or 10 each time. Do this until the diskette drive can be reliably accessed.

## *Administering physical DOS partitions*

A physical DOS partition is a portion of the fixed disk formatted under DOS and reserved exclusively for DOS files. DOS partitions can be created in a variety of sizes; different versions of DOS support different maximum sizes for partitions. DOS Services supports both the DOS 3.3 partitions, which are limited to 32 Mbytes, and the larger partitions that can be created under DOS 5.0.

DOS allows you to create two categories of partitions. Primary partitions are those from which you can boot the DOS operating system. Extended partitions are non-bootable and may be further subdivided into two or more logical drives. See your DOS documentation for further details. DOS Services supports both primary and extended DOS partitions. Your fixed disk may have just one physical DOS partition (the primary DOS partition); it may have both a primary and an extended DOS partition; or it may have no physical DOS partitions.

If your computer has a primary DOS partition, the partition is automatically available as DOS drive E: whenever you run DOS. If your computer has an extended DOS partition and you want to use logical drives within it, you must attach each logical drive that you want to use to an available DOS Services drive letter. Attaching logical drives is described under "Using Physical DOS Partitions" in the DOS sections in the User's Guide. A physical DOS partition is useful on the SCO system for the following reasons:

- Some copy-protected DOS applications that cannot be installed in the file-system shared by DOS and the UNIX operating system can be successfully installed in a DOS partition.
- You can use files and applications contained in DOS partitions under "raw" DOS. That is, you can use them even if you shut down the UNIX system and boot your computer with DOS alone.

If you do not already have a physical DOS partition and you choose not to create one, you can still use all DOS Services features except drive E:. You can also create and use a virtual DOS partition any time after you install DOS Services without removing or reinstalling any files that are currently on your fixed disk. Virtual DOS partitions offer advantages similar to those of a physical DOS partition. Refer to “Administering Virtual DOS Partitions and Virtual Floppy Disks” later in this chapter for more information.

This section tells you how to create, format, and administer a physical DOS partition.

### *Creating and formatting physical DOS partitions*

---

If you want to have physical DOS partitions on your computer, you should create them at the time you install the UNIX system, before you install DOS Services. Boot standard “raw” DOS and use the DOS **FDISK** utility to create and format DOS partitions. Each DOS partition should have a minimum size of 2.5 Mbytes. Some DOS copy-protection schemes cannot install on a partition smaller than 2.5 Mbytes.

If you do not have a physical DOS partition on your computer and wish to create one, you will have to back up your fixed disk, create the DOS partition, and reinstall your UNIX system and all applications. Refer to the documentation packaged with your computer for further information.

You can also add a second hard disk to your computer and put one or more DOS partitions on it. DOS Services automatically senses these new partitions the next time you reboot and makes them available. You still need to associate each of the new partitions with a drive letter. Refer to the section entitled “Using Physical DOS partitions” in your User’s Guide for more information.

### *Default protection of DOS partitions*

---

The UNIX filesystem protection mechanisms apply in only a limited way to DOS partitions. Because physical DOS partitions are not UNIX filesystems, access to individual DOS files in these partitions is not governed by UNIX system user or group ownership or by UNIX system read, write, or execute permissions,

However, a DOS partition itself is accessed as a UNIX file or device and can be protected like any other UNIX file or device.

Here are the considerations that affect access to and administration of DOS partitions:

- Although multiple users can read files on a given DOS partition at the same time, only one DOS process can write to the primary DOS partition or to a

logical drive in the extended DOS partition at one time. In addition, when any DOS process writes to a DOS partition or logical drive, no other DOS processes can read from or write to that partition until the DOS session that was writing to the partition is terminated.

- DOS partitions contain DOS filesystems and are located on a portion of the fixed disk that is physically distinct from the filesystem shared by DOS and the UNIX operating system.
- DOS partitions are accessed as UNIX devices (sometimes called special files). Each of these devices appears to the UNIX system as a single UNIX file, but can contain any number of DOS files and directories. The device name for the primary DOS partition is */dev/dsk/devicename*. The device file name for any DOS partition (if it exists) may be determined by looking in the file */etc/dosdev*. In that file, the first line under the heading "Real DOS device definition" is the name of the primary partition. The next line shows the name of the extended DOS partition. You can find an example of these lines in the DOS section of the User's Guide.
- When you install DOS Services, the DOS partitions (that is, the UNIX system special device files associated with the partitions) are owned by root and are readable and writable by everyone. To check the ownership and permissions of the primary DOS partition, type:

```
ls -l /dev/dsk/devicename
```

To check the ownership and permissions of the extended DOS partition, type:

```
ls -l /dev/dsk/extended devicename
```

where *devicename* is the name of the special device file associated with your DOS partition(s), as determined by looking in */etc/dosdev*.

For both kinds of partitions, the system shows root as the owner and displays the following permissions:

```
brw-rw-rw-
```

The initial "b" identifies these files as block-type special files (as opposed to character-type). UNIX execute permission for the files is not necessary.

Although owned by root, DOS partitions by default are considered to be public resources that any user can access. All users have read permission so they can inspect the contents of the partitions or execute DOS programs stored there. All users have write permission so they can install DOS applications on the partitions. Write permission is granted to all users for an additional reason: even if users do not need to create files within DOS partitions, some DOS applications require write permission when they run and do not work if run from an unwritable filesystem.

## Changing the protection of DOS partitions

---

If you need to restrict access to a DOS partition, you can use one or both of the following methods:

- Use the DOS **ATTRIB** command to make specific files in the DOS partition read-only (that is, remove DOS write permission).
- Change the UNIX system permission mode of the UNIX system special file associated with the DOS partition to restrict access to the partition.

Using the DOS **ATTRIB** command to make a DOS file read-only reduces the risk of that file being accidentally removed or corrupted. The protection offered by DOS **ATTRIB** is limited, however, because anyone who can access a DOS file can also use **ATTRIB** to make the file writable.

To make a DOS file read-only, use the **ATTRIB** command to assign the read-only (**R**) attribute as follows:

```
E> attrib +r filename
```

When the read-only attribute is assigned, the file cannot be deleted or changed. To make the file writable again, type:

```
E> attrib -r filename
```

If you want to restrict access to a DOS partition in a more general way, you can use the UNIX command **chmod** to assign any desired permission mode. For example, to deny access to users outside the group the primary DOS partition belongs to, you could type:

```
chmod o-rw /dev/dsk/devicename
```

For the extended DOS partition, you would type:

```
chmod o-rw extended devicename
```

where *devicename* is the special device file associated with the your DOS partitions.

**NOTE** Changing the permissions for the extended partition changes permissions for all logical drives within that partition.

## *Administering virtual DOS partitions and virtual floppy disks*

---

A physical DOS partition is a section of the fixed disk that is formatted under DOS and reserved for DOS files. DOS Services also allows you to create virtual DOS partitions. A virtual DOS partition is actually a UNIX file that has been formatted as a DOS volume and is used to store DOS files. A virtual DOS partition has advantages similar to those of a physical DOS partition, as well as some special features.

Just as with a physical DOS partition, you can use a virtual partition to install copy-protected DOS applications that cannot be installed on the shared filesystem. Virtual partitions have the following additional advantages:

- You can create a virtual DOS partition easily at any time after you install DOS Services. You can therefore avoid the time-consuming process of reconfiguring your fixed disk to create a physical DOS partition if your system does not already have one.
- Users can create and own virtual partitions. By using personal virtual partitions, they can avoid ownership and permission mode conflicts that may arise from sharing physical DOS partitions.

Because virtual DOS partitions are actually UNIX files and therefore part of the UNIX filesystem (unlike physical DOS partitions):

- Virtual partitions are backed up or restored using standard UNIX backup utilities, in contrast to physical DOS partitions which are backed up with DOS utilities.
- You cannot access a virtual partition when you have shut DOS Services down and booted standard DOS.
- Virtual partitions take up space in the UNIX filesystem.
- Virtual partitions are limited to 7 Mbytes.

After you create a virtual partition, you can attach it as a DOS drive.

DOS Services also allows you to create a UNIX file and format it under DOS for use as a virtual DOS floppy drive. You can use virtual floppies much as you would use physical DOS diskette drive A: or B:.

The following sections show how to create and administer virtual DOS partitions and how to create virtual floppies.

### Using dosadmin to create a virtual DOS partition

---

To create a virtual DOS partition, use the following **dosadmin** procedure:

1. Start **dosadmin** by typing:

**\$ dosadmin**

(You may start **dosadmin** from either the DOS or UNIX command line.)  
The **dosadmin** "DOS Administration" main menu appears.

2. Press the Left or Right Arrow key (<←> or <→>) to move the highlighted field to "DOS Services".
3. Press the Up or Down Arrow key (<↑> or <↓>) to move the highlighted field to the "Create Virtual Partition" field. Press <Enter> to select this item and display the following screen:

The screenshot shows a terminal window with a menu-driven interface. At the top is a title bar with 'File Edit Options' and 'Help'. Below it is a menu bar with 'DOS Administration'. The main area contains a box titled 'Create Virtual DOS Partition'. Inside this box are two input fields: 'DOS Partition Name:' and 'Partition Size:'. Below these fields are two buttons: '< Cancel >' and '<< Create >>'. At the bottom of the terminal window, there is a status line that reads: 'Enter the name of the DOS partition. 1Help 2Choices 3Restore 4Keys 5Clear 6Truncate 7Left 8Right 9Cancel 10Execute'.

4. In the "DOS Partition Name" field, type the full UNIX path of the file you want to use as your virtual DOS partition. For example, to create a virtual DOS partition named */usr/joe/vpart*, type:

*/usr/joe/vpart*

Then press <Tab> to move the highlight to the "Partition Size" field.

5. Enter the size of the virtual partition, in kilobytes (1024 kilobytes equals 1 Mbyte.) The maximum size is 7 Mbytes; the recommended minimum size is 2.5 Mbytes. For example, to create a 2.5-Mbyte virtual DOS partition, type:

2560

Then press **<Tab>** to move to the "Cancel" field.

6. If you choose to cancel the operation and not create the virtual partition, press **<Enter>** while "Cancel" is highlighted. Otherwise, check your entries for accuracy. If any information on the screen is incorrect, press **<Tab>** to move to the field you want to change and retype the entry. (Press **<F5>** to clear a field completely.)

When the information is correct, press **<Tab>** to highlight the "Create" field. Then press **<Enter>**. DOS Services creates the virtual DOS partition you have selected.

## Using virtual partitions

To use a virtual DOS partition, you must attach it as a drive between E: and I: when you start DOS. For example, to use the `/usr/joe/vpart` DOS partition as drive F:, you would start the DOS environment by typing:

```
$ dos +af:=/usr/joe/vpart
```

For the duration of the DOS environment, `/usr/joe/vpart` is accessible as DOS drive F:. Other DOS Services users (or Joe himself) can simultaneously run other DOS sessions with `/usr/joe/vpart` attached as drive F: (or any other drive between E: and I:) provided no user writes to that drive. Simultaneous access to the same virtual partition is allowed only when all access is read-only. As soon as any DOS process writes to the virtual partition, that process has exclusive control of that virtual partition until it exits. When other processes attempt to access the same virtual partition, they receive the message, "Not ready drive F:"

You can use the `dosopt` command to automatically attach a DOS partition whenever you run DOS, but remember the restrictions concerning access whenever you use a virtual partition. Refer to the descriptions of the `dosopt` command in Chapter 5 and the `+a` option in Chapter 6 for further information.

Note that if you have a primary physical DOS partition, it is available by default as drive E: when you start a DOS process. Only one partition at a time is available as drive E:. If you want to use a virtual partition as drive E:, you can use the command `dos +ae:=pathname` even if you have a physical DOS partition, however, E: no longer references the primary physical DOS partition.



You should not use drives C:, D: or J: to attach virtual partitions. These drives have specific functions under DOS Services. See the User's Guide for more information.

### *Administering virtual DOS partitions*

---

Most of the information covered under "Administering Physical DOS Partitions" in this chapter also applies to virtual DOS partitions. A virtual DOS partition differs from a physical DOS partition in that a virtual DOS partition is a single file within the filesystem shared by DOS and the UNIX operating system. This single file contains a DOS filesystem within itself. A virtual DOS partition is not useful from the UNIX environment because the files within it cannot be accessed from the UNIX environment. However, you can move it around, back it up, or delete it using standard UNIX tools. Deleting it frees up space on the shared filesystem.

### *Using dosadmin to create a virtual DOS floppy*

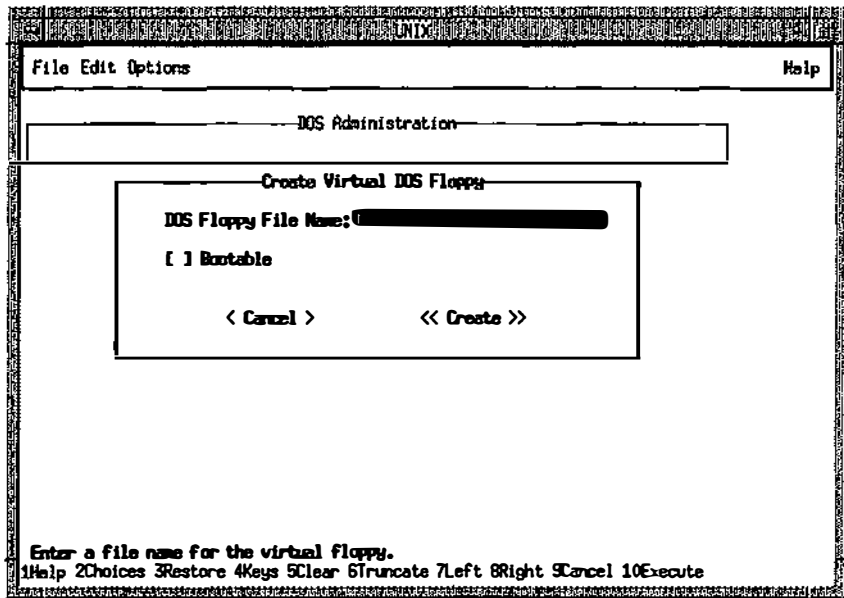
---

A virtual floppy disk is a UNIX file that has been formatted under DOS and contains a DOS volume in a size that corresponds to a standard DOS diskette. You may use a virtual floppy just as you would a physical diskette to store DOS files. You may also make it bootable so that you can start a DOS session from it.

**NOTE** When you boot from a virtual floppy, you normally cannot access the filesystem shared by DOS and the UNIX operating system.

To create a virtual DOS floppy, use the following procedure:

1. Start **dosadmin** by typing:  
**\$ dosadmin**  
(You may start **dosadmin** from either the DOS or UNIX command line.)  
The **dosadmin** "DOS Administration" main menu appears.
2. Press the Left or Right Arrow key (← or →) to move the highlighted field to DOS Services.
3. Press the Up or Down Arrow key (↑ or ↓) to move the highlighted field to the "Create Virtual Floppy" field. Press <Enter> to select this item and display the following screen:



4. Enter the full UNIX pathname of the file you want to use for your virtual DOS floppy in the "DOS Floppy Name" field. For example, to create a virtual DOS floppy named */usr/joe/vflop*, type:

***/usr/joe/vflop***

5. If you want this floppy to be bootable, use the (Tab) key to move to the "Bootable" field and press (Enter) or (Space) to check the selection box next to it.
6. To create the virtual floppy, press (Tab) until the "Create" field is highlighted. Then press (Enter). DOS Services checks the size and density of your physical A: drive. The system creates and formats the virtual floppy to match the size and density of your physical drive. It is made bootable or nonbootable as you have specified.

If you decide to cancel the operation and not create a virtual floppy, press (Tab) until the "Cancel" field is highlighted and then press (Enter).

To use the virtual floppy, you must attach it as floppy drive A: or B: when starting DOS. For example, to use the */usr/joe/vflop* virtual floppy as drive B:, you would start the DOS environment by typing:

***\$ dos +ab:=/usr/joe/vflop***

When you attach a virtual floppy like this, you cannot access a physical DOS diskette drive with the same name (drive B: in this example). You can, however, access physical diskette drives with different names.

You can boot a virtual floppy attached as drive A: if you selected the “Bootable” option when you created the floppy. To boot a virtual floppy, use the **dosboot** command with the **+a** option to attach the virtual floppy to your DOS process. The syntax is:

```
dosboot +aa:= pathname
```

where *pathname* is the full path of the UNIX file containing the virtual floppy. For example:

```
$ dosboot +aa:=/usr/joe/vflop
```

**dosboot** is a synonym for the **dos -l** (“no load image”) option. Refer to Chapter 6 for further information on this command.

## *Installing plug-in cards in your computer*

---

DOS Services allows DOS to use standard hardware devices in a convenient way. Standard DOS hardware devices include displays, disk drives, COM ports, expanded memory (EMS), a mouse, printer ports, and the game port. A few pointers on administering several of these devices appear earlier in this chapter. Also refer to Chapter 6 (especially the description of the **ta** option) for information on using standard devices.

This section tells you how to install, configure, and use DOS hardware devices that are not included as part of DOS Services. Examples of such devices are scanners and LAN cards. DOS Services can use these devices only by communicating directly with them, without the intervention of the UNIX system. This form of device access is called direct attachment.

When you install new cards, the DOS Device Information Popup Window can be helpful. See the DOS sections of the User’s Guide for further information on this utility. When you add, remove, or change computer hardware, you should make new DOS images. See “Making New DOS Images,” later in this chapter, for more information.

## *Using directly attached DOS devices*

---

Install your directly attached DOS device according to the manufacturer’s instructions.

To use a directly attached DOS device, you must issue the **dos** command with the **+a** option. This attaches the device directly to the DOS process. The command structure is complex and you will probably choose not to issue the entire command from the command line every time you use the device. The next section, “Setting up /etc/dosdev,” describes a method for tokenizing the command string. This allows you to issue simplified startup commands that directly attach your device.

The information in this section permits you to test direct attachment of your device from the command line until you have found the combination of options that works best. Typically, you will then tokenize the command string as described in the next section. All of the examples that follow assume that you are operating with root privileges.

The syntax for direct device attachment is:

**+a *device\_specification***

where *device\_specification* consists of the following fields:

|                          |                                        |
|--------------------------|----------------------------------------|
| "io_port_range"          | (typically in the range of 0-3ff, hex) |
| "interrupt_level"        | (0, 2-7, 9-13, or 15)                  |
| "memory_mapped_io_range" | (typically c0000-effff, hex)           |
| "dma_channel"            | (0, 1, 2, 3, 4, 5, 6, 7)               |

The device specification fields must be listed in the order shown, but only those needed must be specified. Fields are separated by dots (.). The first field has a leading dot (between it and +a). When a field is not relevant, a single dot is used. An example of a command that starts the DOS environment and directly attaches a device is:

**# dos +a.2ff5..**

Following are more complete descriptions of each of the device specification fields:

- "io\_port\_range" has this recursive definition:

**X[-Y][,io\_port\_range]**

where the form *X* specifies an I/O port address *X* (in hex) and the form *X-Y* specifies an I/O port address range, from low *X* (in hex) to high *Y* (in hex), that the device uses. As the syntax shows, single or multiple ports or ranges may be given.

- "interrupt\_level" is a single decimal number that denotes which interrupt level is used.
- "memory\_mapped\_io\_range" has this recursive definition:

**X-Y[,memory\_mapped\_io\_range]**

where *X* and *Y* are two hex numbers, separated by a hyphen, denoting the range of the memory-mapped I/O. The first number is the lowest address and the second is the highest address. Each range must start on a 4K boundary and be a multiple of 4K. As the syntax shows, multiple ranges may be specified.

- "dma\_channel" is a single decimal number that denotes which DMA channel is used. Note that devices that use certain DMA modes may not work.

Determine the values for "io\_port\_range", "interrupt\_level", "memory\_mapped\_io\_range", and "dma\_channel" by consulting the hardware technical specifications for the devices being used. If you have difficulty determining the required values for your hardware, contact the hardware manufacturer.<sup>2</sup> The following example shows how to use the +a option to specify a directly attached device, in this case a hypothetical adapter card for a device called "widget."

- The widget card interrupt is 7, the ports range from 240 to 25f, and the device requires an 8K RAM buffer with a starting address of ce000. The device specification is therefore:

```
.240-25f.7.ce000-cffff
```

The last field, DMA channel, is blank because this field is not relevant for this adapter card. You can attach this device to DOS when you start the DOS environment with the command:

```
dos +a.240-25f.7.ce000-cffff
```

If you have an application called "runwidget" that uses this device, you can invoke the application and attach the device at the same time with the command:

```
dos +a.240-25f.7.ce000-cffff runwidget
```

**NOTE** Be sure that the interrupt, the port ranges, and the RAM address you specify do not conflict with any already existing DOS devices (such as the mouse driver or the EMS driver). If a conflict does exist, we suggest you change the settings on your hardware device.<sup>3</sup> This example was specifically chosen to avoid conflicts with other DOS devices. If you try this example but you use settings that do not match those specified here, this example may not work.

Incorrect use of device specifications for directly attached devices can crash the SCO system. Therefore, only the system administrator (logged in as root or super user) is allowed to specify directly attached devices on the command line, as illustrated in the preceding examples. To enable other users to access directly attached devices, you must define tokens for the devices in the *dosdev* file. The next section describes this procedure.

- 
2. You can determine the values currently in use on your system by running the UNIX `hwconfig` command.
  3. If you need to change the default settings for the mouse driver, refer to the file */etc/dosdev*. If you need to change the default settings for the EMS driver, see "Using Expanded Memory (EMS)" in Chapter 8 of this guide.

## Setting up *letcdosdev*

The DOS Services device specification file */etc/dosdev* serves several purposes. *dosdev* includes specifications for several standard devices, including printers, COM ports, and expanded memory. It also includes comments and example “template” device specifications that you might find useful as you set up *dosdev* specifications for other DOS devices.

Normally, you should not alter existing *dosdev* specifications. You can add your own device specifications to *dosdev*, though. The primary reasons for specifying a directly attached device in *dosdev* are:

- To allow DOS Services users to directly attach devices. When you specify a device in *dosdev*, users other than root or super user can directly attach devices.
- To create a short, easily remembered alias (or token) for a device specification. You and other users can use this token with the **dos +a** option instead of fully specifying the device as described in the previous section.

Before specifying a directly attached device in *dosdev*, you should test the device specification using the **+a** option as described in the previous section. Because incorrect use of device specifications for directly attached devices can crash the SCO system, you should conduct your tests in system maintenance mode and at a time when users are not inconvenienced if you need to reboot the system.

The syntax for specifying a directly attached device in *dosdev* is:

*token d device\_specification comments*

Tokens may not contain spaces, commas, or equal signs and may not start with a slash or dot. Tokens also may not resemble a drive letter specification (a single letter followed by a colon). Aside from these restrictions, you can use any alphanumeric characters in tokens. Tokens can be up to 32 characters in length. For example, the *dosdev* line for the directly attached widget card discussed in the previous section might be:

```
widget d .240-25f.7.ce000-cffff. Direct attach widget card.
```

This line can be anywhere in *dosdev*. As long as this line exists in *dosdev*, users can directly attach the widget card with the command:

```
$ dos +awidget
```

Users can also use the **dosopt** command to cause specific applications or the **dos** command to request access to a directly attached device automatically. Use the token defined in *dosdev* when issuing the **dosopt** command. For example:

```
$ dosopt +awidget runwidge
```

When you use **dosopt** like this, DOS Services looks up the specification for the token (“widget” in this example) in *dosdev* when you run DOS, not when you assign the option with **dosopt**. Therefore, if you change the device specification associated with the token in *dosdev*, DOS Services always uses the current specification. Refer to Chapter 5 for further information on **dosopt**.

Note that directly attached devices cannot be shared. As long as one process is using a directly attached device, other processes are prevented from using the device until the process controlling the device exits.

### *Mapping device names*

---

The SCO system allows DOS to access a hardware device via different device specifications than the physical device uses. For instance, assume you have a communications program that uses COM1 by default. If you decide you want the application to use the physical COM2 port instead, you can use the mapping feature to make the application treat COM2 as though it were COM1, without reconfiguring the application or physically moving hardware. This feature is useful whenever you want DOS to view a hardware device as though it were a different physical device.

To make DOS view a hardware device as though it had different specifications, you map the virtual device specification (the specification that DOS sees) to the physical device specification (the actual physical characteristics of the device). The syntax for mapping device names is:

**+avirtual\_device\_specification=physical\_device\_specification**

*virtual\_device\_specification* and *physical\_device\_specification* can be:

- Complete device specifications (including fields for I/O port range, interrupt level, memory-mapped I/O range, and DMA channel).
- Tokens that are defined in *dosdev*.

For example, the tokens **dcom1** and **dcom2** are defined by default in *dosdev*, so you can start the DOS environment and map COM1 to COM2 with the command:

**\$ dos +adcom1=dcom2**

This command causes DOS to treat the physical COM2 port as though it were COM1.

There is no difference in performance when you map virtual and physical DOS devices like this, except for I/O ports, for which it is more efficient to make the virtual and physical port ranges the same.

## Limitations of direct attachment

---

While the foregoing sections describe the steps for attaching devices directly to DOS Services, we cannot guarantee that all devices attached this way will work.

There are many factors that may prevent a device from working under DOS Services. For example, devices that use certain DMA modes may not work. Tape drives are a prime example.

Devices that are highly timing-dependent may fail under the SCO system's multitasking environment. Some devices need to be initialized or reset when the system is booted, and these devices may not work on a DOS invocation. Shared interrupts for devices are not supported on Micro Channel Architectures (MCA) and this may hinder other devices.

Directly attached devices are also sensitive to various configuration changes. Changing the environment — adding more users, changing the load, tuning UNIX parameters, adding or subtracting memory, or any number of other environment changes — can have unpredictable effects on the operation of direct-attached devices.

## Making new DOS images

---

The DOS image is a file that reflects the configuration of your virtual PC environment. The DOS image is a frozen picture, or snapshot, of DOS after it has been booted and loaded into memory and begun running. This image includes information DOS needs about the system configuration, derived from the hardware installed on the system and the contents of the configuration file `/usr/merge/image/config.mki` at the time the image is made. It allows quick startup of a DOS program from the UNIX environment. The default DOS images (for monochrome, CGA, and VGA display adapters) are contained in the following files.

- `/usr/merge/image/std/mono/*.img`,<sup>4</sup>
- `/usr/merge/image/std/cga/*.img`,
- `/usr/merge/image/std/vga/*.img`.

These DOS image files are used by all DOS Services users whenever they run any DOS process unless they specifically request a custom image when they start DOS.

---

4. Hercules display adapters also use the file `/usr/merge/image/std/mono/mono.img`.



There are several reasons to make new DOS images. Typically, when new images are required, the system administrator creates a new default DOS image for each type of display adapter used on the SCO system. The new default images are then available automatically to all DOS users. The following subsections describe when you need to create new DOS images and how to create them. The topics are:

- Hardware Changes that Require New DOS Images
- Changing Special CONFIG.SYS Commands
- Using `mkimg` to Make New DOS Images
- Using `dosadmin` to Make New DOS Images

### *Hardware changes that require new DOS images*

---

The most common reason to make new DOS images is that you have changed your computer hardware. New DOS images are required whenever you make hardware changes that change the BIOS data area or interrupt vectors. You may not know whether a particular hardware change has had these effects. Making new DOS images is simple, however, and it never hurts to make new images even when they are not required. We therefore recommend that you make new default DOS images for all types of displays used on your system after you make any hardware change.

#### *Standard ROMs and on-card ROMs*

When you create a DOS image for a VGA display adapter card, DOS Services uses the data in the display adapter ROM. DOS Services can use either the actual physical ROM on the card while making the image, or a standard ROM supplied (as a file) with the DOS Services software. The images created during the DOS Services installation procedure use standard ROM files, rather than the on-card ROMs, whenever standard ROMs exist. When you make a new image following the initial DOS Services installation, you may be required to choose between a standard ROM and an on-card ROM. In most cases, you should use the standard ROMs. The image-making process is more reliable when you use standard ROMs. The only time to choose on-card ROMs is when you are using a display adapter with non-standard functionality and you want to take advantage of that functionality.

**NOTE** DOS Services may not operate correctly when you use nonstandard video modes. Because extended video modes can vary in unpredictable ways (in their treatment of registers in particular), it is impossible to predict how DOS Services will work when you use these modes.

To create a new DOS image, follow the procedures in “Using `mkimg` to make new DOS images” or in “Using `dosadmin` to make new DOS images,” later in this chapter. “Using `mkimg` to make new DOS images” describes how to use

the command-line utility **mkimg** to make DOS images. “Using **dosadmin** to make new DOS images” describes how to use the **dosadmin** menu utility to make DOS images.

## Changing special CONFIG.SYS commands

---

DOS Services allows you to use nearly all **config.sys** commands as you would on a conventional DOS computer. Commands in the *root* directory **config.sys** file affect all users when they start a DOS process. Commands in a user’s home directory **config.sys** file affect only that user’s DOS processes.

On the SCO system, however, DOS does not interpret certain **config.sys** commands unless they are incorporated into the DOS image you are using. These are **BUFFERS**, **DOS**, **FCBS**, **LASTDRIVE**, **STACKS**, and **SWITCHES**. DOS ignores these commands in the *root* directory and home directory **config.sys** files and any other **config.sys** files you specify with the **+e** option at DOS run-time.

Incorporating **config.sys** commands into your DOS images also enhances startup. If there are certain **config.sys** commands you always execute with every DOS session you start on your system, you should include those in the system’s images. This reduces startup time for all system users.

The factory default DOS images supplied with DOS Services already contain nearly everything you will need. The only alterations you may wish to make are the addition of device drivers and a different value for the **LASTDRIVE** command. After making these alterations you create one or more new DOS images, one for each type of display to which you want the changes to apply. The new images can be made with the **mkimg** utility or with **dosadmin**.

## Adding device drivers to DOS images

---

Never edit the image configuration file **/usr/merge/image/config.mki** directly. This file is used only in the DOS image-creation process. It includes configuration information (such as device driver definitions) that are needed whenever users run DOS processes. The factory default images are made on the basis of the entries in this file. Altering any of its lines is likely to result in unexpected and undesirable DOS Services behavior.

Below **/usr/merge/image/std** are three special directories created specifically for the making new images. There is one for each of the images you might need to make. They are:

- **/usr/merge/image/std/mono**
- **/usr/merge/image/std/cga**

- `/usr/merge/image/std/vga`

In each directory is a file called `CONFIG.SYS` which contains only a single `LASTDRIVE=` line. Add your device driver lines to these `CONFIG.SYS` files and alter the `LASTDRIVE` value if you wish. Copy the DOS device driver files to the appropriate directory for each of the display types to which you want that device added.

The images you make with `mkimg` or `dosadmin` are constructed on the basis of the lines in the image configuration file plus the entries in these `CONFIG.SYS` files.

### *Using mkimg to make new DOS images*

---

You can use the `mkimg` utility instead of `dosadmin` to make new DOS images when you want to make new DOS images for all display cards and you want all images to be made using the standard ROM. To make new DOS images using `mkimg`, follow these steps:

1. Log in as root or become the super user. (Note that you can only use the `mkimg` program from the UNIX command line.)
2. Type:

`# mkimg`

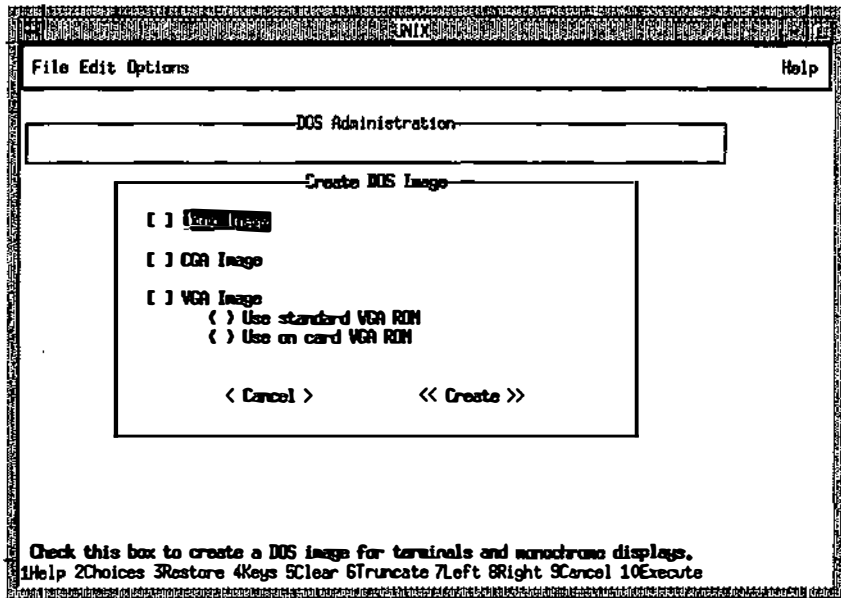
The `mkimg` utility creates new DOS images for all display cards.

### *Using dosadmin to make new DOS images*

---

To make new DOS images using `dosadmin`, follow these steps:

1. Log in as root or become the super user.
2. Start `dosadmin` by typing:  
`# dosadmin`  
(You may start `dosadmin` from either the DOS or UNIX command line.)  
The `dosadmin` "DOS Administration" main menu appears.
3. Press the Left or Right Arrow key (`<<` or `>>`) to move the highlighted field to "DOS Services." A submenu opens under "DOS Services."
4. If the "Create DOS Image" field is not highlighted, press the Up or Down Arrow key (`<↑` or `<↓`) to move the highlighted field to "Create DOS Image." Then press `<Enter>` to select this item and display the following screen:



There are fields for each type of image you can create. These fields vary, depending on your computer hardware. For example, fields for VGA images appear only if you have a VGA card installed.

5. Use the `<Tab>` key to move the highlight from one field to another, and press `<Enter>` or `<Space>` to check the selection box for each image you want to create. You can select any or all of the displayed image types.
6. If your system allows you to create images that use either a standard ROM or an on-card ROM, the `dosadmin` screen offers you a choice between the two types of ROM. The default is to use the standard ROM, so by default this selection has an asterisk next to it. If you want to use the on-card ROM, use the Down Arrow key (`<↓>`) to move the highlight to the "Use on-card ROM" field. Then press `<Enter>` or the `<Space>` to select this field. An asterisk appears between the parentheses.
7. If you decide not to complete this procedure, press `<Tab>` to move the highlight into the "Cancel" field and press `<Enter>`. Otherwise, to complete the procedure, press `<Tab>` until "Create" is highlighted, then press `<Enter>`.

# System files affected by system administration

---

For your reference, the following is a list of DOS Services files affected by common administrative procedures. These system files are used by the general DOS Services user community. Most of these files are described earlier in this chapter. Refer to Chapter 5 in this guide for further information on */etc/dosenv.def*.

- */autoexec.bat*: system-wide AUTOEXEC.BAT file (affects all users)
- */config.sys*: system-wide CONFIG.SYS file (affects all users)
- */etc/dosenv.def*: system-wide DOS environment file (affects all users)
- */usr/merge/image/config.mki*: CONFIG.SYS file used for all DOS images
- */etc/dosdev*: DOS device definitions file
- */usr/merge/image/std/mono/mono.img*: default image for starting DOS with a monochrome or Hercules display adapter
- */usr/merge/image/std/cga/cga.img*: default image for starting DOS with a CGA display adapter
- */usr/merge/image/std/vga/vga.img*: default image for starting DOS with a VGA display adapter

Be sure that you inform all affected users (via email or message of the day) whenever you modify any of these files.

## Chapter 3

# Installing DOS applications

---

This chapter tells you how to install and configure DOS applications on the SCO Open Desktop/SCO Open Server fixed disk, as well as how to remove them. In general, you can install most applications simply by entering the DOS environment as described in the DOS Services chapters in the *User's Guide* and following the application manufacturer's instructions for installing on a fixed disk. You can then configure your application so that it can be invoked directly from the UNIX shell as well as from DOS. This configuration may be done either manually or with the `dosadmin` program. Both procedures are described in this chapter.

A few copy-protected applications cannot be installed on the filesystem shared by DOS and the UNIX operating system. You can install these applications on the DOS partition. For further information on this subject, see "Installing copy-protected programs," later in this chapter.

Some applications cannot be installed on the fixed disk at all because they boot their own operating systems from the distributed diskette to run the program. To run these applications, use the DOS Services `dosboot` command as described in the DOS Services chapters in the *User's Guide*.

Other applications require processor facilities not available in the Virtual 8086 environment created by DOS Services. These can be installed on a real DOS partition, but cannot be configured to run under DOS Services. The sole exception is Microsoft Windows 3.1, for which DOS Services simulates the Protected Mode of 80x86 processors.

This chapter includes the following sections:

- Tips on installing DOS applications to run under DOS Services
- Installing applications with advanced processor requirements
- Configuring DOS applications for use from the UNIX shell
- Installing copy-protected DOS applications
- Removing DOS applications

## *Tips on installing DOS applications to run under DOS Services*

---

The tips offered here assume that you have started a DOS environment under DOS Services and are preparing to install your application according to the manufacturer's instructions.

Note that this discussion assumes that you are installing an application that requires its own subdirectory on the fixed disk. Simple DOS applications or commands (those that have a single .EXE or .COM file and do not require their own subdirectories) can be directly copied into the user's home directory or into the directory C:\USR\LDBIN, rather than requiring an installation procedure.

Before installing an application, decide if the application is for personal use (for use by yourself in your own subdirectory) or for public use (accessible by multiple users and installed in a public subdirectory). It is generally recommended that you install applications for personal use, as the procedure for this type of installation is easier. However, if you have a multiuser system and you are short on disk space, you may prefer to install the application for public use. The next section gives tips on installing an application in a personal directory. The section following that gives tips on installing an application for public use.

### *Installing personal DOS applications*

---

To install a personal DOS application, log into your SCO Open Desktop/SCO Open Server system as yourself, start a DOS session, and follow the normal installation procedures for the application, taking into consideration the suggestions offered below.

Most DOS applications are installed with a program called “install” or “setup” that must be run from an installation diskette. These installation programs generally ask users to specify the following:

- The drive and directory on which the application is to be installed
- The display adapter type
- The printer type
- The mouse type

The installation program may also ask for permission to modify *AUTOEXEC.BAT* and *CONFIG.SYS* files. A few installation programs may offer to configure the application so it can run in 80286 or 80386 protected mode.

Following are some tips on how to respond to the prompts for these items of information.

- **Specifying the drive and directory:** The installation program often displays *C:\* as the default drive and directory. Under DOS Services, *C:\* is the system *root* directory, and since you are not logged in as root, you do not have write permission there. We recommend that you specify a subdirectory under your home directory as the installation location. For example, if your home directory is *C:\USR\PAT* and you are installing Lotus 1-2-3, you might specify *C:\USR\PAT\LOTUS* as the program subdirectory.

If the application does not allow you to specify a subdirectory (that is, the application insists on being installed in the *root* directory of a drive) then specify the *D:* drive.<sup>1</sup> Under DOS Services, *D:\* is automatically substituted to your home directory with the DOS *SUBST* command. (Refer to the MS-DOS manual for a description of the *SUBST* command.)

- **Specifying the display:** Most applications automatically detect the type of display you have. If the one you are installing does not, specify your display adapter type here (for example, *VGA*).

*VGA* display modes are only supported in *X Windows* zoom mode (or on the console when you are not using the *X* environment). If you plan to run programs remotely (on other servers, where zooming is not permitted), if you do not want to zoom, or even when running *X* locally, you may want to specify *CGA* or *HERCULES* as your display adapter.

If you want to run an application in several different graphics modes, you can change the graphics mode by starting DOS with different video images (that is, *dos +acga*, *dos +aherc*, *dos +avga*).

1. Note that if you install an application on the *D:* drive, you may be required to run it from the *D:* drive.



- **Specifying the printer:** Specify LPT1, LPT2, or LPT3 as your printer port. Specify the type of printer attached to your UNIX system as the printer type.

By default, under DOS Services, all DOS print requests to LPT1, LPT2 and LPT3 go to a UNIX print spooler named `doslp`. This print spooler sends its requests to the printer attached to your computer.

- **Specifying the mouse:** The mouse driver provided with DOS Services is especially designed to provide optimum performance in a multitasking environment, and is automatically available to every DOS session you start. Respond "no" if the application asks whether it should install its own mouse driver.
- **AUTOEXEC.BAT and CONFIG.SYS Editing:** Some installation programs ask you if they can modify the standard *AUTOEXEC.BAT* and *CONFIG.SYS* files. Always answer "no" to this question and make any necessary modifications to your personal *AUTOEXEC.BAT* and *CONFIG.SYS* files in your own home directory. Consult the program documentation for a description of the modifications that need to be made.
- **Protected mode options:** If the installation program offers to configure the application for a specific processor, always choose 8086. DOS Services does not support applications that require the features of 80286 or 80386 processors. In particular, DOS Services can not run DOS programs that require protected mode. The one exception is Windows 3.1, which is supported in standard mode. See "Installing applications with advanced processor requirements" later in this chapter.

## *Installing public DOS applications*

---

The tips listed above for installing DOS applications for personal use apply also to applications installed for public use, with the following differences:

- Instead of logging in as yourself, log in as root.
- For the installation directory, specify a subdirectory under *C:\USR\LDBIN*, the system application directory. For example, if you are installing Lotus 1-2-3, you might specify *C:\USR\LDBIN\LOTUS*. If the application prompts only for the drive and not for a directory, specify the J: drive. Under DOS Services, J:\ is automatically substituted to *C:\USR\LDBIN* with the DOS SUBST command. (Refer to the MS-DOS manual for a description of the SUBST command.)
- If the program's installation routine requests permission to modify the system-wide *AUTOEXEC.BAT* and *CONFIG.SYS* files, use your own discretion. Remember that you, as system administrator, are responsible for cleaning up any damage that a poorly written installation routine might do.

Before you let the **INSTALL** routine do any modifications, be sure to make backup copies of both files. When it has done its work, scrutinize the changes carefully by comparing the changed files to the backups.

Your alternative is to determine what changes the installation routine makes and generate those same changes with an editor.

- When you have completed the installation, you may need to use the UNIX **chmod** command to give appropriate read, write, and execute permissions to the application subdirectory and its files so that they are accessible to other users.

## Defining the search path

---

After you have installed a personal or public DOS application, you may want to update the search path. If you have installed a personal DOS application, you can change the UNIX search path that is defined whenever you log in by modifying your home directory *.profile* file. Note that you do not include DOS drive designations (such as C:) in UNIX path definitions. By default, the UNIX search path applies in the DOS environment also. Alternatively, you can set the **PATH** variable in your *AUTOEXEC.BAT* file to define your search path in the DOS environment.

If you have installed a public DOS application, make sure the directory where it is installed is in users' search paths. If necessary, you can update the search path for all users by changing the **PATH** definition in */etc/default/login*. By default, the UNIX search path applies both in the UNIX shell and in the DOS environment. Alternatively, you can set the **PATH** variable in the *root* directory *AUTOEXEC.BAT* file.

Refer to Chapter 8 for additional techniques for managing your search path.

## Installing applications with advanced processor requirements

---

DOS Services creates a two kinds of DOS environment for your DOS software. For Windows 3.1 and its applications, DOS Services creates a Virtual 286 environment which offers access to up to 15 Mbytes of memory. Only Windows 3.1 and its applications can access this feature.

For all other DOS programs, DOS Services creates a 1 Mbyte, DOS environment. This is done by invoking the Virtual 8086 mode of the Intel 80386 or 80486 processor.

Any application which you run in this standard V86 environment must be compatible with the Intel 8086 processor. Any DOS application which absolutely requires an 80286 or later processor does not run under DOS Services. The sole exception is Windows 3.1 and its applications, which are supported in the Standard Mode of the 80286 processor.

For further discussion of memory and how your DOS programs can use it, see "Allocating memory for DOS" in Chapter 5. For more on processor-dependent functions, "Protected mode restrictions" in Chapter 8.

Most DOS applications are meant for the 8086. Of those designed for more powerful processors, many are smart enough to sense the nature of the processor on which they are running and adjust their operation accordingly. For these "intelligent applications" you need to do nothing special. They respond to the V86 environment as an 8086 processor and automatically run in the 8086-compatible "real mode."

For non-Windows applications that offer 286/386 features and fail to adjust to the V86 automatically, there are several ways to compensate:

1. **Install the application for the 8086.** Many DOS applications offer a backward-compatible installation option. The installation routine permits you to specify which processor the application is to run on. By choosing the 8086, your application may run slower or without some of its more advanced features, but it runs under DOS Services.
2. **Choose an earlier version of the program.** If the most current version of the application requires an 80286 or 80386 processor, investigate earlier releases of the software. You may find that an earlier version offers the features you want and is 8086-compatible. For instance, Lotus 123 Version 2.2 runs under DOS Services; Version 3.0 does not.
3. **Install and run under "raw" DOS.** If you have a DOS partition and you are willing to shut down your UNIX system every time you want to run the application, you can install your protected mode application under "raw" DOS. You must shut down the UNIX system, boot "raw" DOS from the DOS partition and install your application. The general procedure is described in an upcoming section.

The same UNIX shutdown is required every time you run the application. Only a single-user system should consider this solution.

## Configuring DOS applications for use from the UNIX shell

---

Once you have installed a DOS application under DOS Services, it is available to you whenever you are using the DOS environment. This section tells you how to configure DOS applications so that you can also invoke them from the UNIX shell. When an application is configured through one of the methods described here, you can invoke it simply by typing its name at the UNIX prompt.

This section describes two methods of configuring DOS applications for use from the UNIX shell: by using the `dosadmin` utility, and manually (using UNIX commands).

### Using `dosadmin` to configure DOS applications

---

Following are step-by-step instructions for using `dosadmin` to configure DOS applications to be used from the UNIX shell. You follow the same procedures whether you are configuring a personal-use application or a public-use application.

To configure an already-installed application using `dosadmin`, follow these procedures:

1. If you are configuring an application for personal use, log in to your system as yourself. If you are configuring an application for public use, log in as root.
2. Type:  

```
$ dosadmin
```

(You may start `dosadmin` from either the DOS or UNIX command line.)  
The `dosadmin` main menu is displayed.
3. The "Applications" submenu should be highlighted. If it is not, press the (Left Arrow) or (Right Arrow) key (← or →) until it is.

### 3. Installing DOS applications

4. Press <Enter> to select "Install". The "Install DOS application for UNIX" menu appears, and your display looks like this:

rothko shell

File Edit Options Help

-----DOS Administration-----

-----Install DOS Application for UNIX-----

Application Directory:  
[Highlighted text box]

DOS Executable File:

< Cancel >                      << Install >>

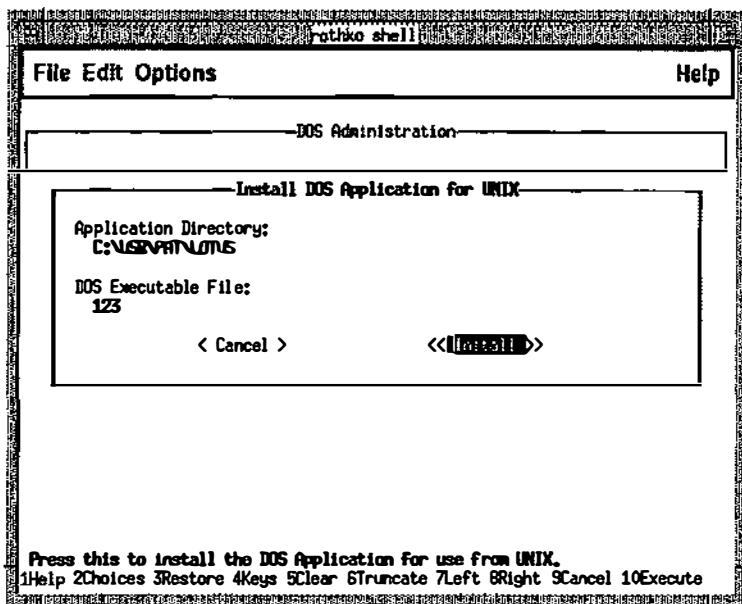
Enter the directory name where the application is installed.  
1Help 2Choices 3Restore 4Keys 5Clear 6Truncate 7Left 8Right 9Cancel 10Execute

5. The "Application Directory" field is highlighted. Type in the full path of the directory where you installed the application. For example, type `C:\USR\PAT\LOTUS`.

Press the <Tab> key to move to the next field.

6. The "DOS Executable File" field is highlighted. Type in the name of the executable DOS program: that is, the name of the command you use when you run the program. It is not necessary to include the filename extension (for example, `.EXE` or `.COM`); `dosadmin` automatically adds the extension if you do not provide it. For example, type `123` if this is the command you use to run the program.
7. Check the fields of the display for accuracy. To make changes, press the <Tab> key repeatedly until the field you want to change is highlighted. Then type in the correct information.

8. When the display shows correct information, press the (Tab) key until "Install" is highlighted. The display should resemble this:



9. Press (Enter) to complete the configuration procedure. An "In Progress" message appears while **dosadmin** configures your application. Then the top-level **dosadmin** menu returns.
10. If you want to configure more executable files, simply repeat steps 3 through 9, providing the appropriate DOS directory name and executable filename at the prompts.

To return to your system prompt, press the (Left Arrow) or (Right Arrow) key (← or →) until "Exit" is highlighted, and press (Enter).

## Manually configuring DOS applications

The previous section described how to make a DOS application executable from the UNIX prompt by using **dosadmin**. Users can manually accomplish the same task by executing a series of UNIX commands. Following are step-by-step instructions for manually performing this configuration. The procedure is illustrated using the DOS program Lotus 1-2-3 as a typical example, and assumes that you want to start the program by typing **123**. (Note that you must perform this procedure from the UNIX shell.)

1. Log in to your system.

2. Change to the directory where the DOS executable program resides.
3. Use the **dosopt** command with the **+y** option to add the default DOS options record to the program:

```
$ dosopt +y 123.exe
```

4. Set permission modes so that the UNIX system recognizes the program as an executable:

```
$ chmod +x 123.exe
```

5. Link the DOS program name to a UNIX system name with no *.bat*, *.com*, or *.exe* extension:

```
$ ln 123.exe 123
```

You can now execute the program from the UNIX prompt by typing its name:

```
$ 123
```

You can perform further configuration with the **dosopt** command. For example, to make 1 Mbyte of EMS memory available every time you run Lotus 1-2-3 from the UNIX prompt, you would execute the command:

```
$ dosopt +aems 123
```

Refer to Chapters 5 and 6 for further information on DOS options and the **dosopt** command.

## *Installing copy-protected DOS applications*

---

A few DOS applications are designed by the manufacturer to prevent installation on more than one machine or execution by more than one user at a time. You can use these applications with SCO systems, but the installation procedures differ according to the type of copy protection used.

Many different copy-protection methods are in use, and it is impossible to recommend installation procedures that apply universally. There are, however, two common classes of copy-protected applications for which we can supply general recommendations that apply in most cases. These classes are:

- applications that use "key disks."
- applications that use special installation utilities.

Procedures for installing applications in each of these classes are described in the following sections.

## *Using a key disk*

---

An application that uses a key disk allows you to install part of the application on the system fixed disk but requires you to insert a protected diskette in the diskette drive whenever you use the application.

Such applications generally require no special installation procedures to work on SCO systems. You can install them by running a DOS environment and following the manufacturer's instructions. When you run an application that requires a key disk, you use the key disk just as you would on a conventional personal computer running DOS.

## *Using special installation procedures*

---

Another class of copy-protected applications uses special installation (and sometimes also removal) procedures to prevent illegal copying of the application. These applications typically include a special installation utility (often invoked with a command such as "install"). They are intended to be installed in their entirety on the system fixed disk so that no key disk is required. You can install such applications on SCO systems but, because different copy-protection methods are in use, you may have to try more than one installation procedure. The procedures, listed in the order you should try them, are:

- Install the application according to the standard instructions under "Installing DOS applications" earlier in this chapter.
- Install the application on a DOS partition (for example drive E:). This procedure is described in the next section, "Installing DOS applications on a DOS partition."
- Shut down your system, boot DOS, and install the application on a DOS partition. This procedure is also described in the next section.

## *Installing DOS applications on a DOS partition*

---

This section describes two methods of installing DOS applications on a DOS partition. These procedures are particularly useful when you install copy-protected applications that cannot be installed on the filesystem shared by DOS and the UNIX operating system.

### *Installing DOS applications under SCO systems*

This procedure is required for a few copy-protected DOS applications that must be installed on an actual DOS filesystem rather than on the filesystem shared by DOS and the UNIX operating system. This installation method works only if you have a DOS partition on your system. A DOS partition can be a physical partition (that can be booted and run independently of SCO



systems), or it can be a virtual partition. If your system does not have a DOS partition, you can create one as described in Chapter 2 of this guide.

**NOTE** These instructions assume you are installing an application on drive E:, which is automatically attached when you start a DOS environment. If you are installing on the extended DOS partition or on a virtual DOS partition, you must first attach the partition to your DOS process. This procedure is described in “Using physical DOS partitions” and “Using virtual DOS partitions and virtual floppy disks” in Appendix B of the DOS Services chapters in the *User's Guide*.

If you are using a drive other than drive E:, substitute that drive letter when performing the steps below.

1. Log in to your system.
2. Start the DOS environment by typing (at the UNIX prompt):  

```
$ dos
```

Your prompt changes to the DOS prompt C:.
3. Change your working drive to drive E: by typing:  

```
C: e:
```
4. Make sure you are working in the directory in which you want to install the application. If necessary, use the DOS DS I (change directory) command to move to the desired directory.  

If you plan to install an application in a physical DOS partition, you must have read and write permission for that partition. (By default, all users have read and write permission on physical DOS partitions.)
5. Install the DOS application according to the manufacturer's instructions. For example, if the application has an installation program named **INSTALL**, invoke it by entering (at your DOS prompt):

**E: install**

(If the manufacturer provides no other installation instructions, use the DOS COPY command to transfer all files from drive A: to drive E:.)

The application is now usable from DOS drive E:.

### ***Installing applications under “raw” DOS***

Use this procedure for the few copy-protected DOS applications with special timing requirements that make them impossible to install while DOS Services is running.

**NOTE** This method works only if your system has a physical DOS partition that can run DOS as a stand-alone operating system.

Since you boot your system under DOS during this procedure, you must have one of the following:

- A bootable DOS system diskette. If you do not already have such a diskette, you can create one by inserting a diskette into drive A: and typing:

**\$ dos format a: -s**

- A bootable DOS partition. (It is bootable if you formatted it using the **FORMAT /S** option.)

Follow these procedures:

1. Shut down the UNIX system according to the instructions in Administrator's Guide.
2. Reboot the system from a bootable DOS diskette or from a DOS partition.
3. Follow the manufacturer's instructions to install the application on a DOS partition.
4. When you are finished installing the application, reboot the UNIX system.

The application should now be usable like any other application on a DOS partition.

Some DOS applications must be installed and run on the same drive. They make a record of the drive letter on which they are installed and fail if you try to run them from any other drive. When you run raw DOS, DOS usually identifies the DOS partition as Drive C:. When you run under DOS Services, that same partition is referenced as Drive E:. The application therefore fails when run from DOS Services.

You can avert this situation by making the application believe it is being installed on Drive E: Thereafter, when you run it from E:, it behaves properly.

After booting raw DOS, use the DOS **SUBST** command as follows:

**SUBST E: C:\**

Then follow the manufacturer's instructions for installing the application on drive E:

When the installation is complete, return the drive to its normal naming conventions with the following command:

**SUBST E: /D**

Each time you run this application, it must believe it is running on Drive E: When you run it from DOS Services the drive is automatically referenced as E:. Any time you run it from raw DOS, however, you must once again evoke

the **SUBST** command shown above. You can do that in a batch file, which then runs the application; or you can do it in the *AUTOEXEC.BAT* on the DOS partition. Which method you choose depends on the requirements of the other applications installed on that partition.

## *Removing DOS applications*

---

Following is an outline of the procedures you follow to remove installed DOS applications from the SCO Open Desktop/SCO Open Server fixed disk. Each of these steps is described in the following sections.

- Remove the application according to the manufacturer's instructions.
- Remove UNIX links to DOS executable files.
- Redefine the search path, if necessary.

## *Removing the application from the fixed disk*

---

Remove a DOS application from the fixed disk by following the manufacturer's instructions. If no instructions are provided, it is usually safe to remove the application's files using the UNIX **rm** or DOS **DEL** command.

Note that some copy-protected applications require special removal procedures. If you do not follow the required procedure, you may not be able to reinstall the application at a future time.

## *Deleting UNIX links*

---

If the DOS application you have removed was configured for execution from the UNIX shell, there are UNIX files that were linked to DOS executable files. (For example, *123* would be linked to *123.COM*.) When you remove DOS executable files (files with extensions ending in *.COM*, *.EXE*, or *.BAT*), the corresponding UNIX links are not automatically removed. When you remove a DOS application, you should also delete any UNIX links to the DOS files you are removing. To delete these links, you simply use the UNIX **rm** or DOS **DEL** command. For example:

```
$ rm 123
```

## *Redefining the search path*

---

After you have removed a DOS application, you may want to redefine your search path. Depending on whether the application you removed was a public or a personal application and on how you define your search path, you might want to modify the **PATH** definitions in */etc/default/login*, your home directory *.profile* file, or an *AUTOEXEC.BAT* file.

## Chapter 4

# Tuning applications with the merge command

---

The **merge** command allows you to tune the operation of the DOS process for certain applications. You use the **merge** command to set the following characteristics of the DOS environment from within the environment:

- You can set the clock interrupt rate that DOS Services uses.
- You can keep programs that poll the keyboard from being put to sleep.
- You can configure the DOS environment so DOS interprets virtual drives or DOS file handles as being either local or remote.

The syntax of the **merge** command is:

```
merge set fastclk on | off
merge set pollsleep on | off
merge set drive local | remote
merge set handle local | remote
merge set netbios on | off
merge set 5f02 on | off
merge set lpt_binary on | off
merge display [option]
merge return option
```

These forms of the **merge** command are described below.

### *fastclk option*

---

The **merge fastclk** option determines whether DOS Services uses the default clock interrupt rate or a lower interrupt rate. The syntax for this option is:

**merge set fastclk on | off**

The DOS Services default is **fastclk on**.

By default, DOS Services uses a clock interrupt rate of approximately 18 per second for DOS, which is the same as a conventional DOS computer. You can improve the performance of most DOS commands and applications by using the **fastclk off** option to reduce the interrupt rate to one per second. To reduce the interrupt rate for your current DOS session, type:

**C: merge set fastclk off**

Some DOS applications sensitive to the clock interrupt rate do not run correctly at the lower rate. If you run one of these applications with **fastclk** set to **off**, you may notice symptoms such as:

- The application runs much more slowly than expected.
- The application hangs.

If you experience one of these problems and it is caused by the interrupt rate, you can correct it by using the **merge** command to set **fastclk** back to **on**.

To set **fastclk on**, type:

**C: merge set fastclk on**

### *pollsleep option*

---

The **merge pollsleep** option determines whether programs that poll the keyboard are put to sleep when they are idle. The syntax for this option is:

**merge set pollsleep on | off**

The DOS Services default is **pollsleep on**.

DOS applications that poll the keyboard can consume system resources even when they are idle by entering a polling loop. Applications that enter a polling loop include (but are not limited to) Lotus® 1-2-3®, dBASE III®, and WordPerfect®.

If not compensated for, these applications reduce system performance because when they poll the keyboard, less CPU time is available to other concurrently running processes. If you run more than one such application at once, overall system response can degrade to an unacceptable level. By default, DOS Services corrects this problem by putting most applications that enter a polling loop to sleep until there is keyboard input.

Some applications may appear to be in a polling loop when they actually are not. By default, the **pollsleep** feature might put these applications to sleep inappropriately, which would not adversely affect other processes, but would cause these applications to perform poorly. This situation is not common, but if you encounter it, you might want to disable the **pollsleep** feature.

To disable this feature, type at the DOS prompt:

**C: merge set pollsleep off**

## *merge network control options*

---

The **merge** command **drive**, **handle**, **netbios**, and **5f02** options all control how DOS Services behaves when DOS applications issue specific system calls usually associated with network transactions. These system calls test whether DOS files or drives are local or remote, whether NetBIOS is available, and whether other network conditions are true or false. DOS Services itself does not include any special support for any particular PC networks. From the point of view of some DOS applications, however, the DOS Services filesystem shared by DOS and the UNIX operating system is a “network” device. These applications use various DOS system calls to determine how (or whether) they should operate. An application may enable or disable file locking, for example, based on its tests of network status.

The four **merge** command options described in the following sections all have defaults that allow DOS applications to run correctly when they access the shared file system. No known applications are adversely affected by the default values for these options. Although unlikely, you may discover an application that does not behave as expected unless you change the value of one or more of these options. We cannot predict specific symptoms that might require changing the value of one of these options because different DOS applications issue network system calls for different reasons.

### *drive option*

Some DOS applications detect whether a DOS drive is local or remote. These applications may not operate correctly if a DOS drive is local when the application expects it to be remote or vice versa.

To configure your DOS environment so DOS interprets DOS Services drives that refer to the shared filesystem (drives C:, D:, and J:, for example) to be local or remote, use the **merge** command with the following syntax:

**merge set drive local | remote**

The DOS Services default is **drive remote**. To make DOS interpret the filesystem shared by DOS the UNIX system as local, type:

**C: merge set drive local**

The **merge set drive local** command causes DOS applications that use DOS interrupt 21H system call function 44H (IOCTL) code 09H (Is Logical Device Local or Remote) to interpret DOS Services drives that refer to the shared filesystem (such as C:, D:, or J:) as local drives. The **merge set drive remote** command causes DOS applications using the same system call to interpret the same DOS Services drives as remote drives. Use these **merge** commands when necessary to configure your DOS environment so applications that are sensitive to local and remote drives work properly.

The following section describes the related **set handle** option.

### *handle option*

Some DOS applications detect whether a DOS file is local or remote. These applications may not operate correctly if a DOS file is local when the application expects it to be remote or vice versa.

To configure your DOS environment so DOS interprets file handles for files in the shared filesystem to be local or remote, use the **merge** command with the following syntax:

**merge set handle local | remote**

The DOS Services default is **handle remote**. To make DOS interpret files in the shared filesystem as local, type:

**C: merge set handle local**

The **merge set handle local** command causes DOS applications that use DOS interrupt 21H system call function 44H (IOCTL) code 0AH (Is Redirected Handle) to interpret files in the shared filesystem as local files. The **merge set handle remote** command causes DOS applications using the same system call to interpret files in the filesystem shared by DOS and the UNIX system as remote files. Use these **merge** commands when necessary to configure your DOS environment so applications that are sensitive to local and remote files work properly.

### *netbios option*

The **netbios** option determines how DOS Services responds when a DOS program asks whether NetBIOS is available. The syntax for this option is:

**merge set netbios on | off**

**netbios on**, the default, causes DOS Services to answer yes when a DOS program asks whether NetBIOS support is available. Note that DOS Services does not actually support NetBIOS; it only responds this way to the initial query from the application.

There are no programs known to be adversely affected by this configuration. However, if you find that a DOS program behaves in an unexpected way that you think may be attributable to this configuration, you may want to turn off netbios as follows:

**C: merge set netbios off**

### ***5f02 option***

The 5f02 option determines how DOS Services treats DOS interrupt 21H system call function 5F02 (Get Redirector List Entry). The syntax is:

**merge set 5f02 on | off**

**5f02 on**, the default, causes DOS Services to enable processing of function 5F02, which is used by the Microsoft redirector.

There are no programs known to be adversely affected by this configuration. However, if you find that a DOS program behaves in an unexpected way that you think may be attributable to this configuration, you may want to disable processing of function 5F02 as follows:

**C: merge set 5f02 off**

### ***lpt\_binary option***

---

The **lpt\_binary** option determines whether DOS Services opens character devices in binary mode or in character mode. **lpt\_binary on**, the default, causes DOS Services to open DOS character devices in binary mode. This configuration allows you to embed the DOS end-of-file character (Ctrl)Z in a file and treat it as an ordinary character. For example, if you print a file containing one or more (Ctrl)Z characters, DOS Services by default treats those characters just like any other characters. A conventional DOS computer would treat (Ctrl)Z as the end of the file and halt printing at the first (Ctrl)Z.

If you want DOS Services to open character devices in character mode instead of binary mode, use the following **merge** command:

**C: merge set lpt\_binary off**

This setting causes DOS Services to treat character devices such as printers the same way conventional DOS does.



## *display option*

---

The **merge display** command displays the current settings for all the **set** options. The syntax for this command is:

**merge display [option]**

For example, to see the current setting for **fastclk**, type at the DOS prompt:

**C: merge display fastclk**

[1] displays the current setting for **fastclk**. To see the current settings for all **merge** command options, type:

**C: merge display**

## *return option*

---

The **merge return** option sets the DOS exit code to a numeric value (0 or 1) that corresponds to the state of the specified option. The syntax is:

**merge return option**

With this form of the **merge** command, the system assigns the value 1 to be "on" or "remote" and 0 to be "off" or "local." For example, if **fastclk** is "on" and you type:

**C: merge return fastclk**

the DOS exit code is set to the value "1". You can use the DOS batch file **ERRORLEVEL** command to check the **merge return** exit code and then take appropriate action. For example, you might include the following lines in a batch file to determine whether **fastclk** is on:

```
merge return fastclk
if errorlevel == 1 goto ISON
echo FASTCLK is OFF
goto END
:ISON
echo FASTCLK is ON
:END
```

## *Using batch files for merge commands*

---

Instead of manually typing a **merge** command to set the value of **fastclk**, **pollsleep**, **drive**, or **handle** when you enter the DOS environment, you can include the appropriate command or commands in a batch file that is executed when you start DOS. In general, you should *not* include **merge** commands in your home directory **AUTOEXEC.BAT** file unless you want to run **merge** commands every time you run DOS. Instead, follow these procedures:

1. Create a batch file, in any convenient directory, that includes the **merge** commands you want to run. For example, you could create a file in your home directory called *SPECIAL.BAT* that includes the commands:

```
merge set drive local
merge set pollsleep off
```

2. When you want to run a DOS application that requires these **merge** commands, start the DOS environment with the command:

```
$ dos +pfile
```

where *file* is the name of the batch file. For example:

```
$ dos +p/usr/wayne/special.bat
```

The **merge** commands in *SPECIAL.BAT* are then effective for the duration of your DOS environment (or until you change them by issuing another **merge** command).

You can also tailor applications to run specific batch files automatically whenever you start them from the UNIX shell. Refer to Chapter 5 and to the description of the **±p** option in Chapter 6 for further information.

## Chapter 5

# *Tailoring the operation of DOS*

---

When DOS Services is installed, it has the characteristics of DOS on a conventional personal computer with 1 Mbyte of memory. With this configuration, you can use DOS alone, use the DOS Services system as a standard UNIX computer, or combine DOS and UNIX functions. Depending on the way you use DOS Services and on the specific DOS commands and applications you use, you might prefer to configure DOS differently. Different DOS applications, for example, require different amounts of memory.

If you use applications requiring large amounts of memory, you can clear nearly all of DOS conventional (low) memory for their use.

This chapter and Chapter 6 (DOS options) tell you how to modify DOS characteristics to suit your needs. The DOS Services system administrator can use the procedures described in these chapters to define systemwide default DOS characteristics. Individual DOS Services users can use the same procedures to define their own preferred DOS characteristics.

This chapter uses simple examples to explain the concepts and procedures you need to understand in order to modify DOS characteristics. Refer to Chapter 6 for complete descriptions of all the DOS options that you can use to customize the operation of DOS. This chapter is organized as follows:

- **DOS defaults:** How DOS defaults affect DOS environments and DOS applications.
- **Using DOS options on the dos command line:** How to override DOS defaults temporarily by specifying options together with the dos command.
- **Installing DOS options to change defaults:** How the system administrator can change DOS defaults.

- **Installing DOS options to override system defaults:** How you can configure commands, applications, and the DOS environment to use your preferred DOS options automatically.
- **Using dosadmin to install and remove DOS options:** How to use the dosadmin menu system to install and remove DOS options.
- **Using dosopt to install and remove DOS options:** How to use the dosopt command to install and remove DOS options.
- **Allocating memory for DOS:** How to configure the DOS environment for the memory needs of individual programs.

## DOS defaults

---

When running under DOS Services, DOS has many characteristics that you can customize to suit your needs. Each characteristic has a default value that you can use or change. Each DOS option described in Chapter 6 controls a specific DOS characteristic. Some of the characteristics you can control are:

- Amount of memory allocated for DOS.
- Which *AUTOEXEC.BAT* and *CONFIG.SYS* files are interpreted when you start DOS.
- Which hardware devices DOS is allowed to access.
- The initial working drive when you start DOS.

Refer to Chapter 6 for more information on specific DOS characteristics, including default values. Unless you override these defaults, they apply to:

- Any DOS environment that any user starts with the **dos** command.
- All DOS commands and applications run from the UNIX shell.

## Using DOS options on the dos command line

---

Whenever you want to change a default DOS characteristic for a single execution of the DOS environment or a DOS application, use command-line options. The syntax for specifying options on the **dos** command line is:

**dos options [command [arguments]]**

For example, to create a DOS environment for Windows 3.1 with 3 Mbytes of memory, type:

**\$ dos +m3**

See “Allocating memory for DOS” at the end of this chapter, for a discussion of how this memory is used. Similarly, to run WordStar with 2 Mbytes of expanded memory to create a file called **report**, type:

```
$ dos +aems2 ws report
```

Options consist of a single alphabetic character (**m** in the previous examples) preceded by either a plus or minus sign (+ or -). The plus or minus sign generally turns the characteristic controlled by that option “on” or “off.” The **+t** option, for example, means “turn on automatic translation of DOS switch characters and path separators,” while **-t** means “turn it off.”

Some options require or are allowed to take parameters. The number “2” in the command **dos +m2**, for example, is a parameter to the **+m** option that specifies the number of Mbytes of memory to allocate.

Options that do not take parameters can be grouped together following a single plus or minus:

```
$ dos +bc -tx compute
```

Options that can take parameters must be specified separately:

```
$ dos +e/usr/dave/config.sys +l/usr/dave
```

The order of options is immaterial. The following two examples are both valid and both have the same effect:

```
$ dos -e -s
```

```
$ dos -s -e
```

DOS options that you specify on the **dos** command line override DOS defaults and also override any DOS options that are installed in an application or options file.

## *Installing DOS options to change defaults*

The system administrator can redefine DOS defaults by installing DOS options in the DOS Services options file */etc/dosenv.def*. This file controls the default characteristics of DOS environments that users start with the **dos** command.

There are two ways the system administrator can change DOS defaults:

- Use the **dosadmin** menu system, which provides convenient ways to change default values for the DOS characteristics that most often need changing.
- Use the **dosopt** command, which is less convenient but more flexible than **dosadmin**.

Both **dosadmin** and **dosopt** are described more thoroughly later in this chapter.

### *Installing DOS options to override system defaults*

---

You can override DOS defaults by installing DOS options in personal options files, DOS commands, and DOS applications. You should install DOS options to override system defaults whenever you want to override a default value more permanently than you can by using DOS options on the **dos** command line. Installed options remain effective until you change or override them. Command-line options are effective only for one execution of a DOS environment or application.

Users can install DOS options in their personal options file *dosenv.def*, which must be in their home directories. This personal options file does not exist until users create it. **dosadmin** creates personal options files automatically when required. Also see “Creating a home directory options file,” later in this chapter.

Like the analogous file in the */etc* directory, the home directory *dosenv.def* controls the characteristics of DOS environments.

The values defined in this home directory file are effective only for the user who owns the file. Values defined in home directory files override values defined in the options files in */etc*. Home directory options files are fully specified. That is, values for every configurable DOS characteristic are defined.

You can also configure specific DOS applications (Lotus 1-2-3, for example) so they use your preferred values when you run them from the UNIX shell. When you install specific DOS options directly in executable DOS command and application files, the options override DOS defaults and options installed in options files. They can be superseded only by using DOS options on the **dos** command line. You do not need to fully specify all DOS characteristics when you install options in a command or application. Unspecified characteristics take their values from the home directory *dosenv.def*, if it exists, or from */etc/dosenv.def* if it does not.

Procedures for installing options in personal options files or in applications are generally the same as the procedures for changing defaults in */etc/dosenv.def*. You can use the **dosadmin** menu system or the **dosopt** command as described in the following sections.

## Using dosadmin to install and remove DOS options

---

The **dosadmin** menu system provides an easy way to change the values of three important DOS characteristics: memory, DOS startup files, and DOS device files. The following paragraphs tell you more about these characteristics and how to modify them using **dosadmin**.

### Characteristics you can tailor using dosadmin

---

You can use **dosadmin** to install DOS options that affect memory, DOS startup files, and DOS device files. Depending on how you use **dosadmin**, you can tailor these characteristics for:

- The DOS environment and all DOS commands and applications run from the UNIX shell.
- Specific DOS commands and applications.

#### *Startup memory*

The DOS Services factory default for DOS memory is 1 Mbyte.

This total memory must be shared, however, by DOS, the application, all the device drivers in use, and any terminate-and-stay-resident (TSR) programs you have loaded. DOS Services uses about 50K bytes for its own driver. For further information, see "Allocating memory for DOS," later in this chapter.

DOS Services runs more efficiently if you allocate only as much memory as you need. For example, if you normally need only 1 Mbyte of memory while in the DOS environment, you can install the 1 Mbyte value as your preferred memory value.

#### *DOS startup file*

On a conventional DOS computer, if you have a file called `\AUTOEXEC.BAT` on your system disk, the commands in that file are executed automatically whenever you boot the computer. When you use DOS Services, you do not boot DOS the same way you would on a stand-alone personal computer. DOS Services, however, can execute an `AUTOEXEC.BAT` file when you use the DOS environment or run a DOS program, just as though you had booted DOS on a conventional personal computer.

The DOS Services factory default is to execute *two* `AUTOEXEC.BAT` files in succession when you start DOS by typing `dos` or when you run DOS programs from the UNIX shell. These two files are the *root* directory `AUTOEXEC.BAT` and your home directory `AUTOEXEC.BAT`.

Using **dosadmin**, you can specify a single DOS batch file that is to be executed instead of the default files when you start the DOS environment or a DOS program, or specify that *no* batch file is to be executed. The file you specify can have any legal DOS or UNIX filename, and you can specify it either by full UNIX path name, for example:

*/usr/joe/dbin/appl.bat*

or by DOS drive and full DOS path name, for example:

*c:\usr\joe\dbin\appl.bat*

### **DOS device file**

On a conventional DOS computer, the file `\CONFIG.SYS` is interpreted automatically when you boot DOS. DOS Services by default treats `CONFIG.SYS` files similarly to `AUTOEXEC.BAT` files: it interprets the *root* directory `CONFIG.SYS` file and your home directory `CONFIG.SYS` file in succession whenever you run the DOS environment or a DOS application.

Using **dosadmin**, you can specify *one or more* configuration files that are to be interpreted instead of the defaults, or you can specify that *no* configuration file is to be interpreted. The files you specify can have any legal UNIX filename. You can specify files either by full UNIX path, for example:

*/usr/joe/dbin/config.sys*

or by DOS drive and full DOS path name, for example:

*c:\usr\joe\dbin\config.sys*

To specify multiple files, list them in the order they are to be interpreted, separated by a space, a comma, or both. For example:

*c:\config.sys, d:\special.sys*

## **Using dosadmin to tailor DOS**

---

**NOTE** When you use **dosadmin** to tailor DOS, DOS Services maintains a record of your configuration in the file *dosenv.def*. This file is stored in your home directory if you are a normal system user or in */etc* if you are the system administrator logged in as root. You do not need to be concerned with the contents of this file, but be sure you do not accidentally delete it.

To use **dosadmin** to tailor DOS, follow these procedures:

- Select the appropriate Configuration menu.
- Identify the DOS application you are tailoring, if you are tailoring a specific application.
- Fill in the fields for memory, startup file, and device file.



The following paragraphs describe these steps in more detail.

## Selecting the configuration menu

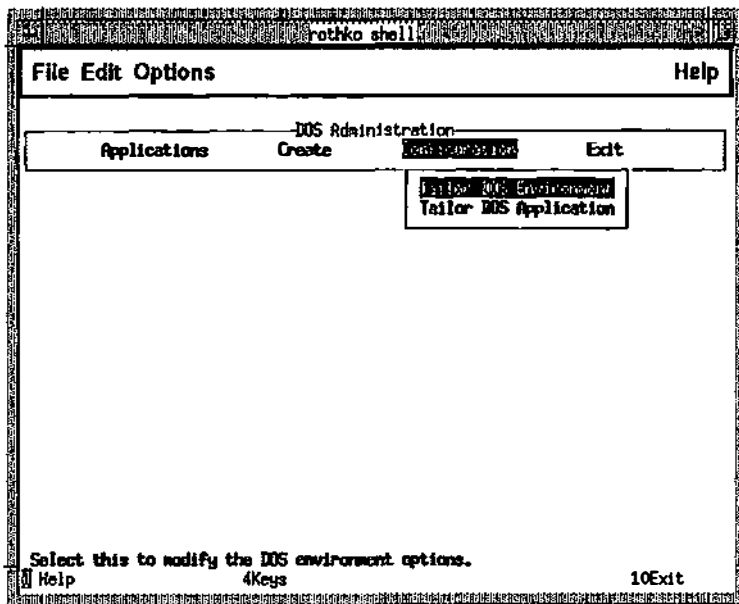
Follow these steps to select the appropriate Configuration menu:

1. Type:

**\$ dosadmin**

(You can type the **dosadmin** command either at the UNIX shell prompt or in the DOS environment.) The main DOS Administration menu is displayed.

2. Press the left or right arrow key (← or →) to move the highlighted field at the top of the screen to "Configuration." Your display looks like this:



The Configuration menu offers two selections:

- Tailor DOS Environment
- Tailor DOS Application

Use the up or down arrow key (↑ or ↓) to highlight the appropriate field and press (Enter). Follow these guidelines to make your selection:

- **Tailor DOS Environment:** Use this selection when you want to modify the characteristics of the DOS environment (that is, the DOS characteristics that apply when you type **dos** and the characteristics of DOS commands and applications when they are run from the UNIX shell). If you are logged in as root, the characteristics you specify *replace* existing DOS

defaults for all users. If you are not logged in as root, the characteristics you specify *override* existing DOS defaults.

You can override characteristics specified with this menu by using DOS command-line options when you use the `dos` command as described earlier in this chapter.

- **Tailor DOS Application:** Use this selection when you want to modify the characteristics of a specific DOS command or application when it is run from the UNIX shell. If you are logged in as root, you can tailor any command or application, although normally you use this menu as root in order to tailor publicly accessible applications in directories such as `/usr/sbin`. You can tailor only commands and applications for which you have write permission.

If you have selected the Tailor DOS Application menu, you must identify the command or application you want to tailor as described in the next section. If you have selected the Tailor DOS Environment menu, skip ahead to “Filling in the configuration menu.”

### *Selecting the DOS command or application*

If you have selected the Tailor DOS Application menu, your display resembles this:

The screenshot shows a terminal window titled "rothko shell". The menu structure is as follows:

- File Edit Options Help
- DOS Administration
  - Tailor DOS Applications
    - DOS Application File Name:  
[Redacted]
    - Startup Memory:
    - DOS Startup File:
    - DOS Device File:
    - < Cancel >      << Change >>

At the bottom of the window, a status line reads: "Enter the DOS executable file name. 1Help 2Choices 3Restore 4Keys 5Clear 6Truncate 7Left 8Right 9Cancel 10Execute"

You must now fill in the “DOS application file name” field to identify the command or application you want to tailor.

Type in the name of a DOS executable file. You can name the file by full UNIX path or by DOS drive and full DOS path, for example:

`/usr/ldbin/lotus.com`

or

`c:\usr\ldbin\lotus.com`

Continue with the steps described in the next section.

### *Filling in the configuration menu*

The Tailor DOS Environment and Tailor DOS Application menus are similar. They both have fields named "Startup Memory," "DOS Startup File," and "DOS Device File." The display resembles this:

rothko shell

File Edit Options Help

DOS Administration

Tailor DOS Environment

DOS Environment Configuration

Startup Memory: 1

DOS Startup File: d:\autoexec.bat

DOS Device File: c:\config.sys, d:\config.sys

< Cancel > << Change >>

Enter the amount of memory for this configuration.

1Help 2Choices 3Restore 4Keys 5Clear 6Truncate 7Left 8Right 9Cancel 10Execute

The values shown for "Startup Memory," "DOS Startup File," and "DOS Device File" are the current values that apply to the DOS environment, application, or applications you are configuring. The actual values shown on your display may differ from those in this illustration.

To change any of the displayed values, press the (Tab) key until the field you want to change is highlighted. Then type in the value you want. If you make a mistake, press the (Bksp) key (←) and retype your entry or press the (F5) key to clear the field. Then retype your entry.

When the “DOS Startup File” or “DOS Device File” field is highlighted, you can use the (F5) key to clear the field, which results in no startup file or device file being interpreted when you start the DOS environment.

Follow these guidelines as you fill in each field:

- **Startup Memory:** The value displayed here is the amount of memory (in Mbytes) reserved for this application when you run it from the UNIX system shell. Note that because DOS Services takes up some space, there is actually somewhat less memory available than the displayed value. “Allocating memory for DOS,” at the end of this chapter, discusses this issue in further detail.

Not all applications need the full amount of memory. If your application runs correctly with less memory, you can increase DOS Services efficiency by specifying a value smaller than 640. Check the application manufacturer’s instructions for the recommended amount of memory.

**WARNING** Specifying too little memory may cause your DOS environment or application to fail.

- **DOS Startup File:** If this field is filled in *only* with *d:\autoexec.bat*, it means *both* the *root* directory *AUTOEXEC.BAT* and personal, home-directory *AUTOEXEC.BAT* files are executed.<sup>1</sup> Note that the home directory *AUTOEXEC.BAT* is executed second, so settings in that file override settings in the *root* directory *AUTOEXEC.BAT*. If you want only one *AUTOEXEC.BAT* file to be interpreted, identify it using DOS drive C: or a UNIX-style path name. For example, filling in this field with *c:\usr\greg\autoexec.bat* causes only Greg’s home-directory *AUTOEXEC.BAT* to be executed. Filling in *\autoexec.bat* causes only the *root* directory *AUTOEXEC.BAT* to be executed.

You are not restricted to files named *AUTOEXEC.BAT* for this field. You can identify any single DOS file, in any directory in the filesystem shared by DOS and the UNIX system, provided that it contains legal DOS batch commands.

For example, if you want your application to run the file *C:\USR\PAT\LOTUS\APPL.BAT* every time you invoke it, type the following in this field:

*C:\usr\pat\lotus\appl.bat*

If you do not want any file to run automatically when you use this application, erase the field by pressing the (F5) key.

---

1. If you are logged in as root when you use DOS, however, only one *AUTOEXEC.BAT* file is executed — the one in the *root* directory.

- **DOS Device File:** This field identifies the device configuration files that you want DOS to interpret when you run your application from the UNIX shell. It normally shows both *c:\config.sys* and *d:\config.sys*. When these two filenames are displayed, DOS interprets both the *root* directory *CONFIG.SYS* and a home-directory *CONFIG.SYS*. If you are logged in as root when you use DOS, however, only one *CONFIG.SYS* file is executed—the one in the *root* directory. (Note that home directory *CONFIG.SYS* files do not exist until users create them.)

You can change or delete the files displayed in this field. For most purposes, however, it is preferable to let DOS interpret the default *CONFIG.SYS* files.

You are not restricted to files named *CONFIG.SYS* for this field. You can identify one or more DOS files, in any directories in the shared filesystem, that contain legal DOS *CONFIG.SYS* commands.

When you have filled in the configuration menu, check your entries for accuracy. If you choose to cancel the tailoring operation, press the (Tab) key until the "<Cancel>" field is highlighted and press (Enter).

When you have entered the values you want, press the (Tab) key until the "<<Change>>" field is highlighted and press (Enter) to save the new values you have specified.

The top-level DOS Administration menu reappears. Press the left or right arrow key until "Exit" is highlighted, and press (Enter). Your UNIX or DOS system prompt returns.

## *Using dosopt to install and remove DOS options*

---

You use the **dosopt** command (like you use the **dosadmin** menu system) to assign DOS characteristics such as memory, startup file, and device configuration file to applications or to your environment. The **dosopt** command is more flexible than **dosadmin**. You can use it to assign many characteristics besides memory, startup file, and configuration file. You can also use **dosopt** where you cannot use **dosadmin**—inside UNIX shell scripts, for example.

As with **dosadmin**, you can use the **dosopt** command to install DOS options that determine:

- Characteristics of DOS environments and applications run from the UNIX shell for all users. Characteristics of specific DOS applications run from the UNIX shell.

The syntax for **dosopt** is:

**dosopt options options\_file**

or

**dosopt options application**

As described earlier, under “Using DOS options on the dos command line,” *options* are key letters preceded by a plus or minus (+ or -) sign that identify the values of specific DOS characteristics. Chapter 6 describes each DOS option thoroughly.

*options files* are files that hold records of DOS options you have installed. In the command form **dosopt options application**, *application* is the name of a specific DOS application—“123,” for example—in which you want to install DOS options.

The names of the options files and the DOS processes they affect are:

- **/etc/dosenv.def**: This file defines system default characteristics for DOS environments and DOS applications started from the UNIX prompt by all users. DOS options defined in this file can be overridden by using DOS options on the **dos** command line or by installing DOS options in home directory *dosenv.def* files.
- **\$HOME/dosenv.def**: A user’s home directory *dosenv.def* file overrides *all* values defined in */etc/dosenv.def* and defines characteristics of DOS environments and DOS applications started from the UNIX prompt by that user. DOS options defined in *\$HOME/dosenv.def* can be overridden by using **dos** command-line options.
- **A specific DOS application**: Although not an options file in the same sense as the files listed above, any DOS executable file can hold a record of DOS options that apply to that file when you run it from the UNIX shell. Options installed in DOS commands and applications override */etc/dosenv.def* and *\$HOME/dosenv.def*. You can override options installed in commands and applications by using **dos** command-line options.

Only the system administrator, logged in as root, can change options in */etc/dosenv.def*, and only with **dosopt**. The file cannot be edited directly with **edlin**, **vi** or any ordinary text editor.

All users can create *dosenv.def* files in their home directories and install options in them. Anyone who has write permission for a command or application can install options in the executable file.

## Creating a home directory options file

---

The system default options file */etc/dosenv.def* is created automatically when you install DOS Services. Home directory *dosenv.def* files, however, do not exist until you create them. There are two ways to create this home directory file:

- Use the **dosadmin** menu system as described earlier in this chapter to modify the characteristics of the DOS environment or DOS applications. If you are logged in as a user who has a home directory, **dosadmin** automatically creates *\$HOME/dosenv.def* the first time you use the “Tailor DOS environment” menu to tailor the DOS environment.
- Copy */etc/dosenv.def* to your home directory and make sure it is writable. If you are working in your home directory, you can use the following commands:

```
$ cp /etc/dosenv.def dosenv.def
$ chmod u+w dosenv.def dosapp.def
```

## Installing DOS options

---

Following are a few examples of **dosopt** commands to illustrate how you install options in options files and in applications. For more information on the meaning of specific options, refer to Chapter 6. All of the examples which follow assume that you already have a *\$HOME/dosenv.def* file, created by one of the methods above.

If you are working in your home directory, the command:

```
$ dosopt +m2 dosenv.def
```

overrides the default values for memory (+m) so that you get 2 Mbytes of memory. If the system administrator enters the following similar command:

```
dosopt +m2 /etc/dosenv.def
```

these values become the defaults for all users who run DOS environments.

Preceding a DOS option with a minus sign (-) generally turns off that function. For example, to prevent *AUTOEXEC.BAT* from running, type:

```
$ dosopt -p dosenv.def
```

## 5. Tailoring the operation of DOS

You can change values at different times without affecting values you specified previously:

```
$ dosopt -p dosenv.def
$ dosopt +t dosenv.def
```

In this example, *AUTOEXEC.BAT* is disabled and automatic path separator translation is turned on in two separate commands. The *AUTOEXEC.BAT* file remains disabled unless it is later explicitly enabled with the **+p** option. If you are working in your home directory, the command:

```
$ dosopt -s dosenv.def
```

changes the default to disable the DOS printer timeout.

The command:

```
$ dosopt +b /usr/vicky/dbin/cl
```

identifies the *cl* program as stream-oriented.

Note that DOS applications that you can run from the UNIX shell normally have two names: a DOS filename with an extension (*.exe*, *.com*, or *.bat*) and a UNIX name without an extension, which is linked to the DOS name. For example, the file *cl.exe* would be linked to *cl*. When you use **dosopt** to install an option in an application, you can use either name. That is, the following two commands have the same effect since *cl.exe* and *cl* are two names for the same file:

```
$ dosopt +b cl
$ dosopt +b cl.exe
```

### Displaying installed options

---

You can display the **dos** options currently in effect by typing:

```
$ dosopt +v
```

This command displays the contents of your home directory *dosenv.def* file, if you have one. Otherwise it displays the contents of */etc/dosenv.def*. The display is similar to the following:

```
File: /usr/vicky/dosenv.def
+ae:=/dev/dsk/0s1 +amouse -b +dc +e +l +m640 +p +s15 +t
```

To display the options installed in a specific options file, use the command:

```
$ dosopt +v file
```

For example, to display the current system default DOS options, issue the command:

```
$ dosopt +v /etc/dosenv.def
```



## Removing installed options

---

To remove an option installed in a command or application, use the **dosopt +z** option. For example, if you have assigned 2 Mbytes of memory to Windows 3.1 using the command:

```
$ dosopt +m2 win.com
```

you can use the **+z** option to remove the installed memory value and allow the memory value from *\$HOME/dosenv.def* (if it exists) or */etc/dosenv.def* to be effective. The command is:

```
$ dosopt +zm wincom
```

When you use **+z** to remove previously installed **+a** ("attach device") options, you can remove individual **+adevice** options or you can remove *all* **+a** options with a single **dosopt** command. For example, assume you have used the following **dosopt** commands to specify attachment of both COM1 and COM2 for the application named APPL:

```
$ dosopt +acom1 appl
```

```
$ dosopt +acom2 appl
```

To remove the **+acom2** instruction but keep the **+acom1** instruction, type:

```
$ dosopt +zacom2 appl
```

To remove all **+a** options from the file APPL at once, type:

```
$ dosopt +za appl
```

Refer to Chapter 6 for further information on the **+a** option.

To remove *all* installed options in a DOS command or application, use an uppercase **"Z"**:

```
$ dosopt +Z appl
```

This command removes all installed options from **appl** and allows the option values in *\$HOME/dosenv.def* (if it exists) or */etc/dosenv.def* to be effective.

The **+z** and **+Z** options cannot be used on the options files *\$HOME/dosenv.def* or */etc/dosenv.def*.

## Restoring DOS options files

---

If for any reason you want to restore the original contents in your system default DOS options file (*/etc/dosenv.def*), you can copy the corresponding file from the directory */usr/merge*. */usr/merge/dosenv.def* is a permanent copy of */etc/dosenv.def* in the form that exists when you install DOS Services.

## *Allocating memory for DOS*

---

When you start DOS or any DOS application, DOS Services allocates 1 Mbyte as the default amount of RAM for DOS use. This is known as “base memory” or “conventional DOS memory.” Base memory is the only area most DOS programs can use for both the program code and the program’s data. Unless special care is taken to conserve this scarce resource, much of this base memory can be absorbed in “overhead.”

DOS 5.0 offers considerably better tools for managing memory than previous versions of DOS, and DOS Services uses those tools to optimize the default amount of memory left for your DOS programs. The factory default DOS images enable the High Memory Area and the Upper Memory Blocks for storage of DOS, device drivers and TSRs. DOS itself is loaded into the HMA.

The DOS Services Bridge driver is the standard redirector that permits DOS programs to access the filesystem shared by DOS and the UNIX system. This driver absorbs about 50K bytes and it too is loaded into the Upper Memory Blocks.

The amount of Conventional Memory available for your DOS applications varies depending on how much additional memory is consumed by the resident portions of DOS and *COMMAND.COM* and by the other drivers and TSRs you load. An average arrangement, consisting of the factory default images with a mouse driver added, leaves approximately 610K of working RAM for your programs. Your own total will vary according to what you load and the portion of memory to which you allocate it. We advise you to read carefully those sections of your DOS documentation that deal with memory management and to use the tools DOS provides, especially the *MEM* command. Load device drivers and TSRs high whenever possible.

### *Running RAM-hungry applications in base memory*

---

Applications that cannot access Extended or Expanded memory and that need a lot of low DOS memory need special handling. We advise that, in addition to loading all possible drivers and TSRs into high memory, you configure your DOS environment to start such programs with only the absolute essentials. A minimal DOS environment includes only the drivers, environment variables, TSRs, and other memory-consuming resources that are essential to run a specific application. All drivers and TSRs that are not critical to the operation of the program are omitted.

To run an application in a minimal DOS environment, you create alternative *AUTOEXEC.BAT* and *CONFIG.SYS* startup files that are customized to meet the minimal requirements of that application. Then you can use any of the following procedures:

- Configure the application so it always uses the alternative startup files whenever users run it from the UNIX shell.
- Start the application with the *dos* command from the UNIX command line, using the *dos +e* and *+p* options to specify your customized startup files.
- Start a DOS environment using the *+p* and *+e* options to specify your customized startup files. Then start the application from the DOS prompt.

The following sections describe these procedures.

### *Preparing minimal startup files*

To prepare minimal startup files for a specific application, simply create an *AUTOEXEC.BAT* file and a *CONFIG.SYS* file that include only the startup commands required by that application. Do not modify your personal home directory *AUTOEXEC.BAT* or *CONFIG.SYS* files or the system default *AUTOEXEC.BAT* and *CONFIG.SYS* files in the *root* directory. Instead, create separate startup files for your RAM-hungry application. These files do not have to be named *AUTOEXEC.BAT* and *CONFIG.SYS*. They can have any legal UNIX filename, and they can be stored in any accessible UNIX directory. For simplicity, we recommend that you name them *AUTOEXEC.BAT* and *CONFIG.SYS* and create them in the same directory containing your application.

For example, if your application is named *RAMHOG* and is in the directory */usr/pat/dbin/ramhog*, create both the *AUTOEXEC.BAT* and *CONFIG.SYS* files in */usr/pat/dbin/ramhog*. If *RAMHOG* is a public application that resides, for example, in the directory */usr/sbin/ramhog*, the system administrator can create the required startup files in */usr/sbin/ramhog*.

You can create the startup files in a variety of ways. You can copy your personal startup files or the system default startup files and edit out unnecessary commands; or you can start with new, empty files and manually add only the required commands. If your application does not require either *AUTOEXEC.BAT* or *CONFIG.SYS*, you need not create the unnecessary file.

### *Using minimal startup files*

You use the *dos +e* and *+p* options to specify the customized startup files you want to use with your application. Specify the full path of the *CONFIG.SYS* file with the *+e* option, and specify the full path of the *AUTOEXEC.BAT* file with the *+p* option.

For example, assume again that your application is named **RAMHOG** and your customized startup files are in the directory `/usr/pat/dbin/ramhog`. One way to run **RAMHOG** is with the command:

```
$ dos +e/usr/pat/dbin/config.sys +p/usr/pat/dbin/autoexec.bat ramhog
```

If either of the startup files is not required, use the `-e` or `-p` option as appropriate. For example, if **AUTOEXEC.BAT** is not used, type:

```
$ dos +e/usr/pat/dbin/config.sys -p ramhog
```

Instead of using the `+e` and `+p` options on the command line, you can install them permanently in the application using the procedures explained in "Installing DOS options," earlier in this chapter. The following command causes **RAMHOG** to interpret `/usr/pat/dbin/autoexec.bat` whenever **RAMHOG** is run from the UNIX shell:

```
$ dosopt +p/usr/pat/dbin/autoexec.bat ramhog.exe
```

Refer to Chapter 6 for more information on the `+e` and `+p` options.

## *Allocating memory for windows*

---

DOS Services supports two versions of Windows, and they use memory in different ways. To make additional memory available to Windows 3.0, you must add Expanded Memory. To make additional memory available to Windows 3.1, you should add Extended Memory.

### *Memory for Windows 3.0*

---

Windows 3.0 runs in Real mode, as it would on an 8086 processor. Windows 3.0 can use Base memory plus Expanded Memory. Because EMS memory was originally developed as an add-in adapter, it is treated as if it were a separate device. To allocate memory beyond the 1 Mbyte limit to Windows 3.0, you "attach" that memory to your DOS process.

From the command line, you use the `+a` option of the `dos` command. The following command requests 4 Mbytes of EMS memory and starts a Windows session.

```
$ dos +aems4 win
```

To achieve the same purpose from the X desktop, bring up the DOS Options Dialog Box. Double click on the Windows icon with the middle button of a three-button mouse or with both buttons of a two-button mouse.

In the Memory box, click on EMS. A drop-down list shows the increments of EMS memory you may add. Click one of the selections. Then click the Start button.

Your Windows session begins with the requested amount of memory.

## Memory for Windows 3.1

---

Windows 3.1 runs in Standard or "Protected" Mode on an emulated 80286 processor. Windows 3.1 can use Base Memory, plus Extended Memory. (It *can* use EMS memory as well, but Extended Memory is much more efficient.) Additional memory beyond the 1 Mbyte limit should always be allocated in the form of XMS memory.

When you allocate memory to Windows 3.1, Windows dispenses that memory to its applications according to their demands and their ability to use memory. You do not need to be concerned with what kind of memory is given to which application. For this reason, DOS Services refers to the memory given to Windows 3.1 as "Standard Memory." That means an aggregate of 1 Mbyte of "conventional" DOS RAM, plus as many Mbytes of Extended Memory as you care to add. The upper limit is 15 Mbytes, total.

Standard Memory is not attached as a hardware device. It is simply allocated as "memory," using the +m (add memory) option of the dos command.

From the command line, the following command requests 4 Mbytes of Standard memory and starts a Windows session.

```
$ dos +m4 win
```

To achieve the same purpose from the X desktop, bring up the DOS Options Dialog Box. Double click on the Windows icon with the middle button of a three-button mouse or with both buttons of a two-button mouse.

In the Memory box, click on Standard. A drop-down list shows the 1 Mbyte increments of Standard memory you may add. Click one of the selections. Then click the Start button.

Your Windows session begins with the requested amount of memory.

Note that Extended memory can be used *only* by Windows 3.1 and the applications that run under it. Special modifications have been made to DOS Services to accommodate Windows 3.1. Other DOS programs cannot access memory in this way, even if they can use Extended Memory under "raw" DOS.

## Chapter 6

# DOS options

---

This chapter explains the options you can use to modify the operation of DOS. To use these options effectively, you should be familiar with the procedures for using the **dosopt** and **dos** commands described in Chapter 5. You can use the options described in this chapter with the **dosopt** command to install options in options files (*/etc/dosenv.def* and *\$\$HOME/dosenv.def*) and in DOS applications. You can use the same options on the **dos** command line when you do not want to install permanent values—that is, when you want the options to be in effect only for the remainder of the DOS session.

This chapter contains a table listing the options and summarizing their use.

Following the table are complete descriptions of each option, arranged alphabetically by option name. Each description includes a “Usage” note, showing the possible ways of using that option with **dos** or **dosopt**. Type all boldface letters and words exactly as shown. The symbol  $\pm$  means you can use the option with either a + or a - (usually to turn the indicated value on or off). Bolded and italicized words (like *file*) are generic terms for which you should substitute actual names (such as the actual name of a file on your computer). Brackets surrounding data (such as [*file*]) mean that supplying the bracketed data is optional. A vertical bar (|) means “either or.” Choose one of the separated items and type it as part of the command. For example, **dosopt +v[options\_file | program]** means you may name either an options file or a DOS program when you use the +v option.

Unless otherwise indicated, you can use any option in the following ways:

- With the **dosopt** command to install your preferred values in either of the following options files:
  - */etc/dosenv.def*
  - *\$HOME/dosenv.def*
- With the **dosopt** command to install your preferred values in a DOS application.
- On the **dos** command line.

For example, the usage line:

**+d[drive]**

means any of the following commands are allowed:

```
dosopt +dd /etc/dosenv.def
$ dosopt +dd $HOME/dosenv.def
$ dosopt +dd cl
$ dos +dd
$ dos +dd cl temp.c
```

You can substitute any legal DOS drive letter for the second “d” in +dd in each of these examples. You can also omit the drive letter to indicate the default (drive C:).

Each option description also includes a “Factory default” note that shows the original system default value (defined in */etc/dosenv.def*) when DOS Services is installed.

| Option                      | Meaning                                                                                                                                                                                  |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>+ados_device</b>         | Attach specified DOS device.                                                                                                                                                             |
| <b>-ados_device</b>         | Cancel DOS device attachment.                                                                                                                                                            |
| <b>+b</b>                   | DOS program is stream-oriented.                                                                                                                                                          |
| <b>-b</b>                   | DOS program is display-oriented.                                                                                                                                                         |
| <b>+c</b>                   | Pass command unchanged to <i>command.com</i> <sup>1</sup>                                                                                                                                |
| <b>+d[drive]</b>            | Set initial current drive.                                                                                                                                                               |
| <b>+e[file[,file ...]]</b>  | Interpret one or more <i>files</i> instead of default configuration files. If <i>files</i> are not specified, default <i>/config.sys</i> and <i>\$HOME/config.sys</i> are used.          |
| <b>-e</b>                   | Do not interpret any <i>config.sys</i> when DOS is run.                                                                                                                                  |
| <b>+h</b>                   | Display help text.                                                                                                                                                                       |
| <b>+l[file   directory]</b> | Load DOS image <i>file</i> or an image file from <i>directory</i> instead of default image.                                                                                              |
| <b>-l</b>                   | Do not use a DOS image. Boot from drive A: instead.                                                                                                                                      |
| <b>+mn</b>                  | Allocate <i>n</i> Mbytes of Standard Memory (Base plus Extended).                                                                                                                        |
| <b>+p[file]</b>             | Run <i>/autoexec.bat</i> and <i>\$HOME/autoexec.bat</i> . <i>file</i> , if specified, is the alternative batch file to be run instead of the defaults.                                   |
| <b>-p</b>                   | Do not run any <i>autoexec.bat</i> .                                                                                                                                                     |
| <b>+s[n]</b>                | Print accumulated DOS printer output when no characters have been sent to the printer for <i>n</i> seconds. <i>n</i> is a timeout value between 5 and 3600 seconds (default 15 seconds). |
| <b>-s</b>                   | Do not print DOS printer output until DOS process exits.                                                                                                                                 |
| <b>+t</b>                   | Translate DOS switch characters and path separators in standard way.                                                                                                                     |
| <b>-t</b>                   | Do not translate DOS switch characters and path separators.                                                                                                                              |
| <b>+v</b>                   | "Verbose" mode: display <b>dosopt</b> acknowledgement message. <sup>2</sup>                                                                                                              |
| <b>+y</b>                   | Null option for DOS .COM files that have no other options. <sup>2</sup>                                                                                                                  |
| <b>+x</b>                   | Forces DOS to start up as an X client, even if XMERGE is not set.                                                                                                                        |
| <b>+z</b>                   | Remove an option from an application. <sup>2</sup>                                                                                                                                       |
| <b>+Z</b>                   | Remove all options from an application. <sup>2</sup>                                                                                                                                     |

1. The **+c** command can be used only on the **dos** command line. It cannot be used with **dosopt**.

2. The **+v**, **+y**, **+z**, and **+Z** options can be used only with **dosopt**. They cannot be used on the **dos** command line.



## *Attach devices to DOS*

---

**Usage:** `+adevice`  
`-a[device]`

**Factory default:** Attach default devices automatically

There are several ways you can configure the hardware devices that DOS processes can access. DOS Services automatically manages access to the most important hardware devices, such as the keyboard, the display, and diskette drives. You can use the `+a` option to alter the default DOS Services treatment of some of these devices. There are also several devices that are not available automatically when you use DOS because they are shared with the UNIX system and other DOS processes. You must explicitly request access to these devices using the `+a` option.

### *Default devices*

---

DOS Services automatically makes the following devices available to DOS processes:

- Console and terminal keyboards
- The fixed disk
- Diskette drives A: and B:
- A mouse
- The primary physical DOS partition
- Console and terminal displays

#### *The keyboard*

Communication between DOS processes and console and terminal keyboards is transparent and automatic.

#### *The fixed disk*

The fixed disk filesystem shared by DOS and the UNIX system, including DOS drives C:, D:, and J:, is also available automatically to all DOS processes.

#### *Diskette drives*

Diskette drives A: and B: are available to DOS as long as they are not mounted as UNIX devices or otherwise used by the UNIX system (by a `cpio` process, for example). The diskette drives are assigned on a first-come-first-served basis to any DOS process that attempts to access them whenever they have been unused for five seconds or longer. Drives A: and B: are assigned and released simultaneously.

## *The mouse*

Any mouse that is properly installed and accessible to the UNIX system is also accessible to DOS (provided it is not actively being used by another process at the moment DOS tries to use it). DOS processes always view the mouse as a Microsoft bus mouse. There is a device driver entry in the default DOS Services `/config.sys` file to support DOS communication with a mouse. When you use the MultiScreen™ feature on the console, the mouse automatically communicates with the screen currently displayed.

## *Physical DOS partitions*

The primary physical DOS partition, if your computer has one, is automatically available as drive E: to all DOS processes. If your computer has an extended DOS partition, it must be explicitly attached. For more information, see “Attaching DOS drives,” later in this chapter, and “Administering physical DOS partitions” in Chapter 2.

## *Display adapters*

DOS Services supports standard serial terminals, PC scancode serial terminals, and consoles using standard monochrome (MDA), CGA, VGA, or Hercules display adapters. In normal use, DOS Services automatically sets up your DOS environment to accommodate the type of console or terminal you are using when you run DOS.

You can explicitly specify monochrome, CGA, VGA, or Hercules capabilities using the `+a` option, although there is typically no need to do so. Your display must have at least the capabilities reflected by the option you select—for example, you cannot specify a CGA adapter type if you are using a serial terminal or a console that has an MDA adapter. The following table summarizes the display adapters you can specify on each type of terminal and the resulting capabilities:

| Capability by physical display type |                                  |          |      |      |
|-------------------------------------|----------------------------------|----------|------|------|
| Option                              | Monochrome or<br>Serial Terminal | Hercules | CGA  | VGA  |
| None                                | mono                             | herc     | CGA  | VGA  |
| +amono                              | mono                             | mono     | mono | mono |
| +aherc                              | -                                | herc     | -    | -    |
| +acga                               | -                                | -        | CGA  | CGA  |
| +avga                               | -                                | -        | -    | VGA  |

If you use the MultiScreen feature with either a VGA display and switch away from a screen displaying a high-resolution graphics application, the application is suspended. Applications running in monochrome, CGA, or Hercules mode continue to run even when their screens are not visible.

### Other standard DOS devices

---

The devices described in this section must be explicitly requested using the **+a** option. The specific **+a** syntax that applies to each device is shown together with the device description.

Except for expanded memory (EMS), these devices are available to only one process at a time. If DOS attempts to access a device that is currently in use by another DOS or UNIX process, DOS Services displays an error message and aborts the DOS process.

#### COM ports

Usage: **+acom1**[=*unix\_device*]  
          **+acom2**[=*unix\_device*]  
          **+adcom1**[=*dcom2*]  
          **+adcom2**[=*dcom1*]

You can attach COM ports to DOS processes either directly (**+adcom1** or **+adcom2**) or indirectly (**+acom1** or **+acom2**). When you use direct attachment, DOS controls the COM hardware directly. When you use indirect attachment, DOS Services mediates the communication between DOS and the COM hardware. Refer to Appendix B of the DOS Services section of the User's Guide for further information on the advantages and disadvantages of both forms of attachment.

*unix\_device* is the name of any terminal line in the form **/dev/ttyxx**, where **ttyxx** is a specific name such as **tty1a** or **tty20**. When you attach a COM port to a DOS process, the specified terminal line is accessible to DOS as COM1 or COM2. By default, **+acom1** means **+acom1=/dev/tty1a** and **+acom2** means **+acom2=/dev/tty2a**. If your computer has a compatible<sup>3</sup> multiport serial card, you can attach other terminal lines as COM1 or COM2 by using options such as **+acom2=/dev/tty20**.

Use the option form **+adcomn=dcomn** when you want a DOS application to view one COM port as though it is a different port. For example, the option **+adcom1=dcom2** causes DOS to treat the physical COM2 port as though it is COM1.

---

3. Not all cards are supported. "Smart" serial cards are those which have their own CPU. They usually offer buffering and caching. "Smart" cards are supported only if their drivers have been specially constructed to support DOS Services. Consult your "Smart" card documentation for more information.

## Expanded Memory (EMS)

Usage: **+aems[n]**

EMS memory was originally available only on an adapter board which was installed in the computer. Even though DOS Services emulates expanded memory in a completely different fashion, it is still “attached” to your DOS session as if it were a separate device. For a discussion of how DOS programs can access EMS memory, see “Allocating memory for DOS” in Chapter 5. DOS Services simulates DOS expanded memory as specified by Lotus, Intel, and Microsoft.<sup>4</sup> **+aems** requests the default of one Mbyte of expanded memory. To request a different amount, use **+aemsn**, where *n* is 512, for 512K bytes of expanded memory, or 1, 2, 3, 4, 5, 6, or 8, specifying Mbytes.

There is a device driver entry for EMS in the system default */config.sys* file, so EMS is available only if DOS interprets this file or another *config.sys* file that has a similar entry.

Each DOS process can request its own private expanded memory, so multiple DOS processes and multiple users can use EMS simultaneously.

Refer to Chapter 8 for technical information on the device driver DOS Services uses for expanded memory.

## Printer ports

Usage: **+alpn**

By default, DOS Services sends all DOS printer output to the UNIX print spooler, so DOS does not require access to printer ports. If you want DOS to control a printer port directly, attach the port using the **+alpn** option, where *n* is 0, 1, or 2. Refer to “Administering DOS Printers” in Chapter 2 for further information.

## The game port

Usage: **+agame**

To use the game port with DOS, use the **+agame** option.

4. *The Lotus/Intel/Microsoft Expanded Memory Specification*, Version 4.0, Lotus Development Corporation, Intel Corporation, and Microsoft Corporation.

## Attaching DOS drives

---

Usage: `+adrive_letter=:unix_file_name[e,i,r]`

As mentioned earlier, the filesystem shared by DOS and the UNIX system, including DOS drives C:, D:, and J:, is always available to DOS, and DOS Services automatically manages access to diskette drives.

The primary physical DOS partition (if your computer has one) is accessible as drive E:. The `+a` option `+ae:=/dev/dsk/ devicename`, which is included in the default `/etc/dosenv.def` file, is required to make the DOS partition accessible as drive E:.

The `+ae:=/dev/dsk/devicename` option is one example of a general option form that you can use to access physical DOS partitions, virtual DOS partitions, and virtual DOS floppies. To access any of these devices with DOS, you use the `+a` option and specify a DOS drive and a UNIX filename. The UNIX filename identifies the file that contains the DOS partition or virtual floppy. Follow these guidelines when you select a DOS drive letter:

- Use drive letter a or b to access a virtual floppy.
- If you attach a virtual floppy using either a or b as the drive letter, you cannot simultaneously access a physical diskette drive with the same name. You can, however, access physical and virtual floppies with different names at the same time.
- Use drive letters e, f, g, h and i to access the primary physical DOS partition, logical drives within the extended DOS partition, and virtual DOS partitions. For a list of the devices currently attached, look at `/etc/dosdev`.
- The primary physical DOS partition (`/dev/dsk/devicename`) is accessible as drive E: by default. You can use drive letter e to access a virtual partition or a logical drive within the extended partition, but if you do, the primary physical partition is not simultaneously accessible as drive E:. For details, see "Using physical DOS partitions" in the DOS Services section of the User's Guide. You can also use `+a` with a drive letter other than e (that is, f, g, h, or i) to access `/dev/dsk/devicename` if you choose.

Virtual partitions and floppies have the same limitations as physical DOS partitions. Multiple DOS processes can read them at the same time, but only one process can write to each of these devices at a time. When one process starts writing, other processes are prevented from both reading and writing to that device until the DOS session that owns the original process is terminated.

## *Specifying immediate, exclusive, or read-only access*

You request access to a DOS drive at the start of your DOS session using the `+adrive_letter=unix_file_name` option. DOS Services does *not* immediately attach the specified device to your DOS process. DOS Services attaches the drive only when DOS actually attempts to read or write to the device. The requested device may become unavailable between the start of your DOS session and the time when DOS actually attempts to access it. In this case the DOS process may display an error message or behave incorrectly. To avoid these problems, you can request that the device be attached to your DOS process immediately or exclusively.

To specify immediate attachment, include the *i* key letter following the UNIX filename. The DOS process then aborts immediately if the specified device is unavailable (for example, if another process is writing to it or it does not exist).

The *e* key letter requests exclusive access to the specified device so that no other process can access it. This option is useful whenever you want to prevent other processes from reading or writing to a device while you use it. The *e* key letter automatically implies the *i* key letter. That is, exclusive attachment also means immediate attachment. It is therefore unnecessary to specify *i* explicitly when you use *e*.

The *r* key letter is equivalent to using a write-protect tab on a physical diskette—it prevents you from making any changes to the designated devices. Note that the *r* key letter does not prevent *others* from modifying your files unless you combine it with the *e* key letter.

For example, to set up your DOS environment with the primary DOS partition attached as read-only, you could use a command like this:

```
dos +ae:=/dev/dsk/devicename,r
```

The order of the *i*, *e*, and *r* key letters is immaterial, but each must be preceded by a comma. These key letters can be used only with devices that DOS Services interprets as files—the physical DOS partition, virtual partitions, and virtual floppies.

## *Specifying custom devices*

---

**Usage:** `+acustom_device`

You, your system administrator, or your DOS Services distributor can create device definitions in the file `/etc/dosdev` for hardware devices other than those described in the previous sections. See Chapter 2 for information on setting up and using custom hardware devices, including details on `/etc/dosdev`.

## Canceling device specifications

---

**Usage:** `-a[device_name]`

The `-a` option prevents access to DOS devices when you use it on the `dos` command line or install it in an application. You can also use `-a` to remove device specifications from options files. The `-a` option can affect access only to those devices you can attach with the `+a` option. You cannot use `-a` to affect access to devices such as the diskette drive that are managed automatically by the DOS Services system.

### *Using -a to prevent DOS from accessing devices*

On the `dos` command line, `-a device_name` prevents DOS from attaching the specified device even if there is an installed `+a` option in an options file or application that requests that the device be attached. For example, the command:

```
$ dos -ae:
```

starts the DOS environment without allowing access to the DOS partition. It is not necessary to type `dos -ae:=/dev/dsk/devicename`. You do not have to supply the UNIX pathname of the device when you use `-a` with the name of a DOS partition (either physical or virtual) or a virtual floppy.

When installed in a DOS application, the `-a` option automatically prevents access to specified devices or to all devices. `-a device_name` prevents access to the specified device unless you override the `-a` option with an explicit `+a` option. For example, the command:

```
$ dosopt -acom1 xtalk
```

prevents CROSSTALK from requesting access to the COM1 port unless you specify `+acom1` on the `dos` command line.

Use the `-a` option without a specific device name to deny access to *all* devices. For example, the command:

```
dosopt -a xtalk
```

prevents CROSSTALK from accessing *all* DOS devices. Users can override the installed `-a` by using a `+a` option on the `dos` command line.

Installing the `-a` option in an application to deny access to a device is not the same as *removing* the `+a` option. For example, assume you have installed the `+acom1` option in the application CROSSTALK. The command:

```
$ dosopt -acom1 xtalk
```

*replaces* the `+acom1` request with `-acom1`. Then, when you run CROSSTALK from the UNIX shell, it does not request access to COM1 even if `/etc/dosenv.def` or your home directory `dosenv.def` contain explicit `+acom1` instructions.

Refer to the description of the `+z` option later in this chapter for information on removing options from applications.

### *Using -a to remove defaults from options files*

You can use a command with the following syntax to remove device specifications from options files:

```
dosopt -a[device_name] options_file
```

For example, the command:

```
$ dosopt -acom2 /usr/dave/dosenv.def
```

removes the `+acom2` specification in `/usr/dave/dosenv.def` but does not affect other `+a` options in that file.

When device specifications are removed from `/etc/dosenv.def` files, users can still request access to any device by installing `+a` options in their home directory `dosenv.def` files, by installing `+a` options in their own applications, or by using `+a` on the `dos` command line.

## *Display-oriented and stream-oriented DOS programs*

---

**Usage:** `±b`

**Factory default:** `-b` (display-oriented)

When a DOS program does all its I/O via standard I/O, it is called *stream-oriented*. Most standard DOS commands, such as `DIR`, `SORT`, and `TREE`, are stream-oriented programs. DOS applications that do not write directly to the system screen (many compilers, for example) are also stream-oriented programs.

When a DOS program avoids some or all of the standard I/O mechanisms, it is called *display-oriented*. DOS applications that write directly to the system video memory (including many text-processing, database, and spreadsheet programs, as well as most games) are display-oriented programs.

To properly handle I/O for a DOS program, DOS Services must know whether it is stream-oriented or display-oriented. The standard DOS commands in `/usr/sbin` are assigned the appropriate value (in most cases, stream-oriented) when DOS Services is initially installed. All other DOS programs are assumed to be display-oriented unless you specify otherwise. You can run any DOS program with the default assignment of display-oriented. To take advantage of the additional flexibility available with stream-oriented programs, though, you should identify these programs as such using the `+b` option. When you



identify a stream-oriented DOS program with the **+b** option, you can use UNIX pipe and I/O redirection mechanisms to manipulate the program's input and output.

To determine whether a program is display-oriented or stream-oriented, follow these guidelines:

- Interactive programs like DOS word processors are generally display-oriented and should have the default (**-b**) assignment.
- Any program that requires a graphics display adapter is display-oriented.
- Most noninteractive programs (such as compilers) are stream-oriented.

If you are uncertain whether a noninteractive program is really stream-oriented, you can perform the following test. Start the DOS environment and run the program in question, redirecting the input or output (as appropriate) to a file. For example, to test output redirection for the program **APPL**, issue these commands:

```
$ dos
C: appl > temp
```

When the program has finished executing, display the contents of the file you redirected to (**TEMP** in this example). You can do this with the DOS **TYPE** or UNIX **cat** command:

```
C: type temp
$ cat temp
```

or use a text editor to display the contents of the **TEMP** file. If the output of your DOS program is correctly captured in this file, the program has stream-oriented output.

To find out whether **APPL** accepts stream-oriented input, you can test it with a command such as:

```
C: appl < con
```

to test input redirection from the console. If **APPL** behaves correctly, it accepts stream-oriented input.

**NOTE** If you run a program that is really display-oriented with the **+b** option, screen output is lost. To abort the program, send a UNIX interrupt (by default, the **<Del>** key).

## *Pass command directly to DOS command interpreter*

---

**Usage:** `dos +c command`

**Factory default:** None

DOS Services by default performs the following actions when you run a DOS command from the UNIX shell:

- Searches your UNIX path for the executable program.
- If the executable program is found, interprets any DOS options that are installed in the file.

The `+c` option causes the command to be passed directly to the DOS command interpreter, `command.com`. DOS Services does not search the UNIX path or interpret any DOS options installed in the application when you use `+c`.

The `+c` option is useful with commands that use names (such as DOS device names) that are not meaningful to the UNIX system. For example, the commands:

```
$ dos +c a:wp
$ dos +bc "dir | sort"
```

run Word Perfect from the diskette drive and sort the output of the `DIR` command. DOS Services runs both commands properly even without the `+c` option, but it finds the programs more quickly when the `+c` option is included in the command. Without the `+c` option, DOS Services searches your UNIX search path for files called, literally, "a:wp" or "dir | sort" before passing the command to `command.com`. With the `+c` option, the command is passed immediately, without searching your UNIX path.

Although `+c` prevents DOS Services from interpreting DOS options installed in the application, options files are still interpreted. Any options specified on the `dos` command line are also interpreted.

## Set initial current drive

---

**Usage:** `+d[drive]`

**Factory default:** `+dc`

The `+d` option sets the initial current DOS drive when you enter the DOS environment or run a DOS program. The *drive* parameter can be any drive letter from "a" to "z." For example:

```
$ dos +dd
```

puts you in the DOS environment with drive D: as your current drive. The `+d` option with no *drive* specification causes the default drive (C:) to be used.

The drive you specify with `+d` can usually be any valid drive—a physical floppy, a physical DOS partition, an extended DOS partition, a portion of the UNIX filesystem being used as a virtual floppy or virtual DOS partition—any drive available to DOS. When the `+d` option is used alone, the only prerequisite is that the drive must exist at the time you name it with the `+d` option. If it does not, DOS aborts.

When `+d` is used in combination with the `+p` option to run an *AUTOEXEC.BAT*, the drive must be one of the drives shared by the DOS and UNIX systems. It cannot be a DOS-only drive like E:.

## Interpret configuration file

---

**Usage:** `+e[file[,file ...]]`  
`-e`

**Factory default:** `+e` (interpret */config.sys* and *\$HOME/config.sys*)

Use the `+e` option to tell DOS Services whether or not DOS should interpret a configuration file when it runs. By default (`+e`), DOS interprets two files in succession:

1. The system default *config.sys* file in the *root* directory (*/config.sys*).
2. Your personal home-directory *config.sys* file.

*\$HOME/config.sys* files do not exist until users create them. If one of these files does not exist, DOS Services interprets the file that does exist. When both files exist, DOS Services behaves as though the contents of */config.sys* and the home-directory *config.sys* are all in a single *config.sys* file.

To specify that you want a different file to be interpreted when you run DOS, name the file (by full path name) along with the **+e** option. For example:

```
$ dos +e/usr/dave/dbin/config.sys
```

runs the DOS environment with the configuration specified in */usr/dave/dbin/config.sys* instead of the configuration specified in */config.sys* and *\$HOME/config.sys*. The specified file can be any UNIX file—it need not be called “*config.sys*.” It must, however, contain valid *config.sys* commands.

When you specify two or more files, DOS interprets the contents of the files, in order, when it starts. For example:

```
$ dos +e/usr/ldbin/config.sys,/usr/dave/dbin/config.sys
```

No default *config.sys* files are interpreted automatically when a filename appears following **+e**. Therefore, if you want one or more files interpreted *in addition* to the defaults, you must list all files explicitly:

```
$ dos +e/config.sys,/usr/dave/dbin/config.sys
```

The **-e** option tells DOS not to interpret any configuration file.

## Help text

---

**Usage:** **+h**

**Factory default:** None

Use the **dos +h** or **dosopt +h** command to display a help screen that briefly describes all the available options. You can use the **+h** option only on the **dos** or **dosopt** command line. It cannot be installed in an options file or in an application.

## Alternative DOS load image

---

**Usage:** `+l[file | directory]`  
`-l`

**Factory default:** `+l` (use system default image)

The default DOS images (one each for monochrome, CGA, and VGA display adapters) are contained in the files:

- `/usr/merge/image/std/mono/mono.img`
- `/usr/merge/image/std/cga/cga.img`
- `/usr/merge/image/std/vga/vga.img`<sup>5</sup>

The option `+l` without a file or directory name means the default image. The option `+lfile` uses the specified DOS image file rather than the default DOS image to execute the invoked program. *file* is the full path name of the alternative image, which can have any legal UNIX name. Note that there is no space between the `+l` and the name of the file.

In the `+ldirectory` form, *directory* is the directory that contains the DOS image subdirectory you want to use. When you use this form, DOS Services automatically searches the specified directory for an image file that corresponds to your display. The image files in the specified directory must be named *mono/mono.img*, *cga/cga.img*, or *vga/vga.img*.

When you use VGA images, the directory containing the image must also contain the file or *vgarom.img* in addition to *vga.img*. These files are created automatically when you create the corresponding image file. The procedures for creating DOS images are described in Chapter 2.

It is rarely necessary to store different alternative DOS images and load them with the `+l` option because DOS Services accommodates nearly all configuration changes automatically when DOS is run.

The `-l` option starts a “DOS environment” with no image, booting the virtual machine’s BIOS from the reset address `f000:fff0`. You can use the `-l` option to boot either a physical or virtual diskette. The diskette need not be a DOS diskette as long as it is bootable and is compatible with the virtual personal computer environment you have specified. Many games designed for personal computers, for example, do not run under DOS and need to be booted

---

5. Hercules display adapters also use the `/usr/merge/image/std/mono/mono.img` file.

from the diskette drive using the `-l` option. To use such a game, insert the diskette in the diskette drive and type:

```
$ dos -l
```

If the game is stored in a virtual diskette, you can run it by typing:

```
$ dos +aa:=pathname -l
```

where *pathname* is the name of the virtual diskette. When you run a "DOS" session this way on the console, you can use the UNIX MultiScreen feature to view and work with other concurrent UNIX or DOS sessions. The process booted with the `-l` option, however, typically runs independently of any other UNIX or DOS activity. That is, usually there is no connection between the process invoked with `dos -l` and the shared filesystem, and no interprocess communication between DOS or the UNIX system and the `dos -l` process. To stop a session started with `dos -l`, press `(Ctrl)(Alt)(Del)` or use the KILL DOS terminal control code described in Appendix A. The `dosboot` command is a synonym for `dos -l`.

DOS

## DOS memory size

---

**Usage:** `+mn` **Factory default:** `+m1` Whenever you use DOS on the DOS Services system, the DOS process requests a specific amount of memory for the environment used by the DOS process. Using the `+m` option, you can request a specific amount of Standard Memory (base memory plus extended memory) for a DOS process.

See "Allocating memory for DOS" in Chapter 5, for a discussion of how this memory is used.

To specify explicitly the amount of memory you want, use the form `+mn`, where *n* is an integer between 2 and 16, representing Mbytes. For example:

```
$ dos +m2
```

requests 2 Mbytes for the DOS environment.

Only windows 3.1 can use Standard Memory beyond 1 Mbyte. The command:

```
$ dos +m4 win
```

requests four Mbytes and runs Windows.

Remember that DOS Services overhead consumes a small amount of the memory available to DOS, so the actual amount of memory available to DOS applications is slightly less than the amount of memory that you request. Remember also that specifying too small an amount of memory may cause DOS applications to fail. See *Allocating Memory to DOS* in Chapter 5. for details.

## Run *autoexec.bat*

---

Usage: **+p**[*file*]  
**-p**

Factory default: **+p**

Use the **±p** option to tell DOS Services whether or not DOS should interpret an *autoexec.bat* file when it runs.

Use **+p** alone in options files, in applications, or on the **dos** command line to cause DOS to interpret first */autoexec.bat* and then *\$HOME/autoexec.bat*. Use **-p** to prevent interpretation of any *autoexec.bat* files.

When you use the **+p** option, be sure you do not put a space between the **+p** and the *file*. The **+p** option by itself, with no filename, executes both the systemwide and home directory startup files. If **+p** is followed immediately by *file*, that file is executed as a startup file. But, if **+p** is followed by a space and then *file*, the space sets off *file* as an argument to the **dos** command. DOS Services interprets *file* as the name of a DOS command to be run. The result is that the systemwide and home directory startup files are executed, then *file* is executed as a DOS command, and the DOS environment is terminated.

The */autoexec.bat* and *\$HOME/autoexec.bat* files do not exist until the system administrator or individual users create them. If one of these files does not exist, DOS Services interprets the file that does exist. When both files exist, DOS Services behaves as though the contents of both files are all in a single *autoexec.bat* file. The contents of *\$HOME/autoexec.bat* override the settings made by */autoexec.bat*, because the user's home directory startup file is the second one to be read.

In the form **+pfile**, *file* is the name of a single batch file that DOS is to run. The file can have any legal UNIX name and can be anywhere in the UNIX file-system. You must specify it by its full UNIX path name.

When you run DOS with the **+p** option in effect, a temporary file */tmp/aennn.bat* is created, where *nnn* is the UNIX process ID of the DOS server. This file is normally removed when DOS exits, but if the DOS server is killed, the file is not deleted.

## Change DOS printer timeout

---

Usage: **+s[n]**  
**-s**

**Factory default:** **+s15** (spool to UNIX system with a 15-second timeout)

Both in the DOS environment and from the UNIX prompt, DOS printing by default is handled through the UNIX print spooler. The **±s** option controls when accumulated DOS printer output is sent to the UNIX spooler. By default, DOS printer output is sent to the UNIX spooler when any one of the following events occurs:

1. The program or command exits. "Exit" means the program finishes and your DOS or UNIX prompt (C: or \$) returns. Many commands (such as the DOS PRINT and COPY commands) exit automatically when they are done. Other programs require a specific user action to exit.
2. The amount of time specified by the printer timeout value has elapsed since DOS last printed. The default printer timeout is 15 seconds.
3. DOS receives a BREAK (for example, you press <Ctrl>C at the DOS prompt).

Use the **+sn** option to change the default 15-second timeout. The variable *n* is a timeout between 5 and 3600 seconds. If *n* is omitted, the default value of 15 seconds is used. The **-s** option makes the timeout infinite, which prevents DOS printer output from being sent to the UNIX system until the DOS process exits or you press <Ctrl>C.

The **±s** option has the same effect as the DOS Services **printer** command **/t** option. The **±s** option can be used on the UNIX command line or installed in options files and applications. The **printer** command must be used in the DOS environment. Both **±s** and the **printer /t** option are effective only when DOS printer output is spooled to the UNIX system. When you attach a printer directly to DOS using a command such as **dos +alp1**, the **±s** option and the **printer /t** option have no effect.



## *Translate DOS switch characters and path separators*

---

**Usage:** `±t`

**Factory default:** `+t` (translate switch characters and path separators according to standard DOS Services conventions)

By default (`+t`), DOS Services requires UNIX-style switch characters and path separators when you issue DOS commands from the UNIX shell. For example:

```
$ dir /tmp -w
```

Before passing the command to DOS, DOS Services automatically translates this command to:

```
dir \tmp /w
```

The `-t` option prevents DOS Services from translating switch characters and path separators on a DOS command line. For example, consider a hypothetical DOS command `COMPUTE`, which does simple arithmetic calculations like division and subtraction. With the default translation in effect, a command such as:

```
$ compute 8 - 4
```

would be translated to:

```
compute 8 / 4
```

Translation of `" - "` and `" / "` in contexts like this, where they are not intended to be interpreted as path separators or switch characters, can result in undesired behavior.

To prevent this translation, use the command:

```
$ dos -t compute 8 - 4
```

You can also use the `dosopt` command to install the `-t` option in `COMPUTE` so it is effective each time you run the application from the UNIX shell:

```
$ dosopt -t compute
```

The `-t` option does not prevent the UNIX shell from interpreting metacharacters. If you use metacharacters in a DOS command, follow the guidelines listed in Chapter 8.

## Dosopt “verbose” mode

---

**Usage:** dosopt +v [*options\_file* | *program*]

**Factory default:** None (do not display acknowledgements)

Use the +v option with dosopt to display the DOS options installed in an application or *dosenv.def* file. For example, a typical display in response to the command:

```
$ dosopt +v database
```

is:

```
File: database
*-a *-b **dc *+e **l **m1 *-p +s20 *+t
* defaults from /usr/dave/dosenv.def.
```

This display shows the values of all DOS options used when DATABASE is run. These values include both:

- Values that have been installed explicitly in DATABASE with the dosopt command:

```
$ dosopt +s20 database
```

- Values that are taken from an options file because they are not explicitly installed in DATABASE.

The values taken from the options file are marked with an asterisk (\*), and the options file is listed in the last line of the display. The options file is either the system default options file */etc/dosenv.def* or the user’s home-directory *dosenv.def* file.

You can use the +v option to display the values installed in an options file:

```
$ dosopt +v /usr/dave/dosenv.def
```

You can include the +v option on the same command line as other dosopt options. For example:

```
$ dosopt +v +s20 database
```

This command installs the option +s20. Then it displays all options effective for DATABASE, as described above.

The command dosopt +v without a program or options file name prints the values that apply to the DOS environment—your home-directory *dosenv.def* file, if you have one, or */etc/dosenv.def*, if you do not.

## *Set X Client startup*

---

**Usage:** `+x`

**Factory default:** X Client Operation Off

The `+x` option forces DOS to start up as an X client, even if the `XMERGE` variable has not been set.

## *dosopt null option*

---

**Usage:** `dosopt +y program`

**Factory default:** None

Use the `dosopt +y` option to create a null DOS options record in a DOS application. The `+y` option is only necessary for DOS .COM files that are to be executed from the UNIX shell and that have not already had DOS options installed using the `dosopt` command. The options record that `+y` creates has no effect except that it allows DOS Services to identify a DOS .COM file as a DOS program when you run it from the UNIX shell. For example, assume you want to run the DOS `CALC.COM` program from the UNIX shell. Install the `+y` option by typing:

```
$ dosopt +y calc.com
```

The standard MS-DOS .COM files distributed with DOS Services already have appropriate `dosopt` records, and you do not need to use `dosopt` to change the records unless you want to change the default behavior of these commands.

## *Remove assigned options*

---

**Usage:** `dosopt +zoption program`  
`dosopt +Z program`

**Factory default:** None

Use the `+z` option to remove an option previously assigned to a DOS application with `dosopt`. For example, you might use `dosopt` to identify the DOS program `WORK` as a stream-oriented program requiring 512K bytes of expanded memory and using the DOS configuration file `/usr/dave/config.new`:

```
$ dosopt +b +aems512 +e/usr/dave/config.new work.exe
```

Values other than those assigned explicitly to **WORK** come from an options file—either */etc/dosenv.def* or *\$HOME/dosenv.def*. The command:

**\$ dosopt +zb +ze work.exe**

removes the **+b** and **+e** options from **WORK** and allows the option values installed in the relevant options file to take effect. Each **+z** requires one and only one option letter. If you want to remove more than one option, use **+z** once for each option you are removing, as illustrated above. The command **dosopt +za *application*** removes all **±a** options from *application* if there are multiple **±a** options installed at the time you use the **+z** option. The command **dosopt +zadvice *application*** removes only the **+advice** option without affecting any other **±a** options that might exist in *application*.

The **+Z** option removes *all* installed options from DOS applications. For example, the command:

**\$ dosopt +Z work.exe**

removes the complete record of assigned options from **WORK** and allows all options in the relevant options file to take effect.

You cannot use the **+z** or **+Z** option on options files (*/etc/dosenv.def* or *\$HOME/dosenv.def*).

## Chapter 7

# Using DOS Services NLS features

---

This chapter deals with configuring your DOS Services system for use with languages other than U.S. English. It describes the DOS Services features that support multiple languages in an environment shared by DOS and the UNIX operating system.

This chapter includes the following topics:

- the features offered by DOS Services NLS,
- prerequisites for using NLS features,
- the theory behind National Language Support,
- an explanation of DOS Code Pages and UNIX Code Sets,
- how DOS handles NLS procedures,
- how the UNIX system handles NLS procedures,
- how DOS Services handles NLS procedures,
- sample NLS procedures for VGA displays,
- sample NLS procedures for ISO terminals,
- how to avoid name conflicts in an NLS environment,
- how to print in an NLS environment,
- how to use the utilities provided for text conversion.

## Features

---

As delivered, your system recognizes and displays all the characters necessary for English as written in the U.S. If you use only data in U.S. format, you can skip this chapter entirely. If you use data in other languages, however,

you need to reconfigure the system to recognize characters that are not part of the default character set.

Support of multiple languages is often called *National Language Support* or *NLS*. DOS Services NLS includes the following features:

- Support of multiple DOS code pages. You can select the DOS code page best suited to the language you use.
- Automatic translation of keyboard input and displayed output. For example, you can view a DOS text file that contains non-ASCII characters even if you use an ASCII terminal.
- The text conversion programs `dos2unix`, `unix2dos`, and `charconv`. You can use these programs to convert text from one code page or code set to another and convert from DOS to UNIX format or vice versa.

### *Prerequisites*

---

To take full advantage of DOS Services NLS, you need to run DOS commands and applications on a VGA console, in a VGA DOS environment inside an X window, or on an ISO terminal. ISO terminals conform to standards set forth by the International Standards Organization and support the ISO Latin-1 character set. Digital Equipment Corporation's VT320 is one well-known example.

Always read the NLS sections of your DOS documentation before you try to do NLS procedures. This chapter assumes you have some familiarity with DOS code pages. The main concepts and procedures are described here, but consult your DOS manuals for details. Also see your DOS manuals for more complete information on related DOS commands such as `CHCP`, `COUNTRY`, `KEYB`, `MODE`, and `NLSFUNC`.

Additional useful information about UNIX operating system NLS characteristics appears in the *SCO UNIX System International Operating System Guide* and the *SCO UNIX System Operating System International Supplement: Release and Installation Notes*.

You can use the material in this chapter in several different ways, depending on your level of familiarity with NLS.

- **If you who are already familiar with NLS concepts and functions**, you can skip the Theory section entirely or simply skim it for new information.
- **If you are unfamiliar with NLS procedures**, refer to your DOS documentation first. Get a good understanding of the commands and procedures DOS uses. Then read the Theory section of this chapter, paying special attention to the few differences in the way DOS Services handles character sets.

- **If you are in a hurry to get a system set up for non-English computing, you can skip the Theory section of this chapter temporarily.** Nevertheless, you should read that section when time permits; it presents information that you need for full comprehension.
- **If it does not work as you expect, keep in mind that the procedures described in the second section are intentionally brief.** They make certain assumptions about how your system is set up. If you follow the instructions and your system does not respond as expected, it probably means that your actual setup differs from those assumptions. If that happens, read your DOS documentation and the Theory section for a full understanding of the underlying concepts. Then customize the procedures for your own particular needs.

## *Theory of National Language Support*

---

As delivered, your SCO Open Desktop/SCO Open Server system is configured to recognize the characters of English as written in the United States of America. You can enter all the characters of U.S. English from the keyboard and display all those characters on the screen. If all your computing is done with this kind of data, you do not need to do anything further.

If, however, you want to perform computing functions on data that uses the characters or conventions of other languages, you need to reconfigure the system. The keyboard must be remapped so you can enter non-U.S. characters. The screen needs to be provided with information on how to display non-U.S. characters. Additional data must be provided to account for differences in the conventions used for punctuation, dates, and currency. Both DOS and the UNIX operating system provide standardized functions for this language-related reconfiguration.

U.S. English is written with twenty-six characters. Other European languages use most of these same characters, but often add other glyphs to express sounds not present in English. The majority of these are written as Roman characters with diacritical or accent marks added, but others are derived from Greek.

DOS supports different character sets that fill the needs of most European languages. See the table later in this chapter under "DOS National Language Support codes" for a list of the character sets that DOS Services supports. DOS Services currently does not support any non-European languages such as Japanese.

## *DOS code pages and UNIX operating system code sets*

---

All symbols used by computers are stored and manipulated as numbers. An eight-bit byte of computer data offers 256 distinct bit combinations. All European languages are encoded in relatively few characters—few enough to be encoded in a single byte of computer data.

Both DOS and the UNIX system use tables that determine how the numeric codes stored in the computer are converted into displayed letters, numbers, symbols, and other characters. The DOS tables are called *code pages* and the UNIX tables are called *code sets*.

### *Differences between DOS code pages and UNIX code sets*

---

*Code pages* and *code sets* are essentially just two names for the same type of object: a table that relates 256 written symbols to the 256 bit combinations available in a single byte. Different languages are written with different character sets. Therefore, different code pages and code sets are required to represent different languages.

If you use only DOS or only the UNIX system, you do not need to be aware of any differences between DOS code pages and UNIX code sets. When you use DOS Services to combine DOS and the UNIX system, however, you may notice some differences between the DOS and UNIX environments. Some characters in data and filenames created with one operating system may be displayed as different characters when you view them with the other operating system. For example, suppose you have a DOS file called *MEMO* containing the following text:

This memo describes the features of our æNET product.

If you issue the command `cat memo` from the UNIX shell, the words might be displayed like this:

This memo describes the features of our \*NET product.

Similar character conversions can occur when you view DOS text on different types of terminals.

These character transformations occur because each character is represented by a code number. The same code number may exist in both your current DOS code page and UNIX code set, but it may be matched with a different symbol in each.

Later sections in this chapter explain how to set up your DOS and UNIX environments so you get the most consistent behavior when you combine DOS and the UNIX system. They also describe the DOS Services `dos2unix`,



**unix2dos**, and **charconv** programs, which you can use to convert text files in a variety of different ways.

## How MS-DOS handles NLS

---

These sections cover the concepts and procedures MS-DOS uses to provide National Language Support.

### Varieties of DOS code pages

---

MS-DOS recognizes two kinds of code pages. A *hardware* code page is built into a hardware device. A *software* code page is provided in software form and stored in code page information (*.cp1*) files.

Following are some of the code pages commonly supported by MS-DOS and various hardware devices:

- **Code Page 437** supplies characters for U.S. English.
- **Code Page 850** is called the Basic Multilingual Code Page. It supplies characters that suffice for most of the languages of Europe, North America, and South America.
- **Code Page 852** supplies characters for Slavic languages.
- **Code Page 860** supplies characters for Portuguese.
- **Code Page 863** supplies characters for Canadian French.
- **Code Page 865** supplies characters for Nordic languages.

In the U.S., hardware devices commonly use code page 437 by default. Hardware devices designed for use in other countries use other code pages by default.

Not all devices can recognize all software code pages. Some printers, for instance, can print only the symbols contained in their hardware code pages. Thus, a particular character may be stored in the computer's memory as part of a software code page, but the attached printer may be unable to print that symbol correctly when it is sent. Some DOS monitors are limited in the characters they can display for the same reason. See "Display considerations," later in this chapter, for further details.

Not all Code Pages and Code Sets are supported for every operation. A particular character set may be viewable on the console, for instance, but not viewable at an attached terminal or printable by the printers attached. See the Release Notes for a list of the DOS Code Pages and UNIX Code Sets that DOS Services supports.

### *Code page switching*

---

DOS allows you to work alternately in several languages on the same machine. This is accomplished by *code page switching*. You can work first in the characters of one language, then switch code pages to work in the characters of another language. Each device affected must be prepared ahead of time with the DOS MODE command. The actual switching is done with the CHCP (CHange Code Page) command.

### *Tailoring DOS for different languages*

---

The following list summarizes the procedures required to set up a standard DOS computer for a different character set. Refer to your DOS documentation for more complete information about following these procedures on a conventional DOS computer. Specific examples of these procedures as they apply to the DOS Services environment appear later in this chapter.

- **Configuring for country** is done with the COUNTRY command. This command is placed in the CONFIG.SYS file. It sets country-specific variables, such as the formats used for dates, times, and currencies. It also establishes the character collation sequence.
- **Configuring system devices** is done by placing DEVICE= statements in the CONFIG.SYS file. These statements tell DOS which hardware and software code pages to associate with each device.
- **Loading memory-resident NLS code** is done by placing the NLSFUNC command in the AUTOEXEC.BAT file. This must be done before you try to specify code pages or keyboard codes.
- **Preparing (loading) code pages** is done by placing the MODE *device* CODEPAGE PREPARE command in AUTOEXEC.BAT. It prepares code pages for those devices that support code page switching.
- **Mapping the keyboard** is done with the KEYB command.
- **Switching between code pages** is accomplished with the CHCP command.

### *How the UNIX operating system handles NLS*

---

A UNIX operating system must perform the same basic activities described above for DOS, but the system interface is simpler. Nearly everything is done by defining a single LANG variable, which takes care of setting all other language- and place-related variables. LANG is of the form:

**LANG=language\_territory.codeset**

For instance, LANG=french\_swiss.8859 would set the language to French as spoken in the French-speaking sections of Switzerland, with ISO Latin 1 (ISO

8859-1) used as the character set. All dates, times, and currencies would be expressed as is customary in that region.

There are two different ways that LANG can be defined:

1. In the file */etc/default/lang*. This file contains the LANG definition in the syntax shown above. Although this syntax is characteristic of environment variable definitions, the LANG variable as defined in */etc/default/lang* is not actually an environment variable.<sup>1</sup> The definition in this file applies to all users unless they override it using the second method.
2. As an environment variable. If you define LANG as a UNIX operating system environment variable, this definition overrides the definition in */etc/default/lang*. You can define LANG just as you would define any other UNIX operating system environment variable. For example, if you use the Bourne shell, you can define the LANG environment variable at your shell prompt using the following syntax:

```
$ LANG=language_territory.codeset
$ export LANG
```

In addition, the UNIX operating system collects user messages into *message files*. This simplifies translating the operating system and its applications into a new language. All the message strings are translated into the new tongue and collected into a new message file.

Switching languages under the UNIX system, then, is a simple matter of resetting one variable.

## How DOS Services handles NLS

---

This section describes how DOS Services integrates the characteristics of DOS and UNIX operating system NLS support.

### Display considerations

---

A VGA console can display any character set for which the font files are loaded. The nature of your display, however, limits the availability of automatic character conversion and the other NLS features offered by DOS Services.

---

1. For example, the command `echo $LANG` does not display the LANG definition when LANG is defined only in */etc/default/lang*.

The complete set of DOS Services NLS features is available when you run DOS:

- on an ISO terminal,
- on a VGA console, or
- in a VGA DOS environment inside an X window.

Throughout this chapter, the phrase *VGA display* refers to a VGA console or to a VGA DOS environment inside an X window. The phrase *VGA session* refers to a DOS session run on any of these displays.

Only a subset of DOS Services NLS features is available when you use any of the following displays:

- MDA, CGA, or Hercules consoles,
- standard ASCII terminals
- PC scancode terminals.

On a conventional DOS computer, monochrome, CGA, and Hercules displays are always restricted to DOS code page 437. The same restriction applies to these displays when you use DOS Services. However, you can use the DOS Services `charconv` command to convert text files created with other code pages or code sets to code page 437 before using them with these displays. `charconv` is described near the end of this chapter.

The same restrictions apply to standard ASCII terminals and PC scancode terminals. In addition, there are some DOS characters that cannot be displayed on ASCII terminals. When you use DOS to display text containing these characters, DOS Services replaces them with asterisks.

### *Setting up the DOS Services NLS environment*

---

There are two ways to set up your DOS Services environment for NLS. One uses the DOS *startup files* `AUTOEXEC.BAT` and `CONFIG.SYS`; the other uses *DOS images*. Both require that you also set certain UNIX environment variables.

The “startup files” method is similar to the procedures you would follow on a standard DOS computer, which require adding appropriate commands to your `AUTOEXEC.BAT` and `CONFIG.SYS` files. With this method, as on a standard personal computer, every time you start DOS, these commands are executed and your DOS environment is set up for NLS.

The second method allows you to put DOS NLS commands into appropriate DOS images. A DOS image is a frozen picture, or snapshot, of DOS after it has been loaded into memory and is running. This image includes information DOS needs about the virtual PC configuration. There is a DOS image for every supported display type (VGA, Hercules, CGA, and monochrome). By install-

ing NLS commands and drivers in DOS images instead of in *AUTOEXEC.BAT* and *CONFIG.SYS*, DOS sessions started with these images are automatically configured for an NLS environment.

The advantage of putting NLS commands in *AUTOEXEC.BAT* and *CONFIG.SYS* is its simplicity. By putting a few commands in these startup files, any user can quickly set up an NLS DOS environment. Different users can have different NLS environments by putting NLS commands appropriate for their needs in their personal *AUTOEXEC.BAT* and *CONFIG.SYS* files.

The "startup file" method has several drawbacks, however. Since these NLS commands are executed every time you start DOS, they can slow down the DOS startup time. In addition, while a VGA DOS session can support all NLS commands, a CGA or a monochrome DOS session supports only a limited number of these commands (see the next section). Therefore, every time you start a CGA or monochrome DOS session, you see several NLS-related error messages as NLS commands are executed by your startup files. Although not harmful, these error messages can be distracting.

The second method for setting up NLS (using DOS images) overcomes most of these shortcomings. Since there is a separate image for each type of display, you can put only the appropriate NLS commands in each image. In addition, since an image contains an environment that is already set up, the system spends no time executing these NLS commands every time you start DOS. (The commands have already been executed.) There are two drawbacks to this method. First, it loads the NLS commands permanently into your DOS environment and they can take up precious RAM. Second, they affect all users. In addition, you must be logged in as root to create DOS images.

If you use NLS only on occasion or if only a few users on your system need it, then we recommend the first method. However, if you want to set up system defaults for a particular NLS environment, we recommend the more permanent DOS image method.

Procedures for setting up ISO terminals for multilanguage use are different from the procedures for setting up other displays. A later section of this chapter describes the steps you need to follow if you use an ISO terminal.

## Setting UNIX operating system environment variables

---

DOS Services uses the UNIX operating system environment variables **CODEPAGE**, **COUNTRY**, and **KEYB**, in addition to the **LANG** variable described earlier. The **LANG** definition is used by DOS Services as well as other SCO Open Desktop/SCO Open Server applications. The **CODEPAGE**, **COUNTRY**, and **KEYB** variables exist specifically to support DOS Services NLS functions. System-wide values for these variables are defined in the files */etc/profile* and */etc/cshrc*. They have the following default values:<sup>2</sup>

```
CODEPAGE=pc437
COUNTRY=1
KEYB=us
```

To provide the greatest consistency between the UNIX and DOS environments, defines these three environment variables to correspond to the values you use with the DOS **MODE**, **CHCP**, **COUNTRY**, and **KEYB** commands. Note that the **CODEPAGE** environment variable has “pc” prepended to the code page number. Define **COUNTRY** and **KEYB** using the same values you use in DOS. See the table in the next section for a list of valid country codes, code pages, and keyboard codes.

Users can customize their own NLS environments and the system administrator can change the defaults for all users by redefining these environment variables.

## DOS National Language Support codes

---

The table that follows is a chart of languages (by country) supported by DOS Services. Column 2 gives the code to use with the **COUNTRY** command. Column 3 shows the default code page used for that country and the alternate code page that is usually switched in with the **CHCP** command. Column 4 gives the code issued with **KEYB** to remap the keyboard.

Consult the Release Notes for information about any newly supported languages that are not listed here.

---

2. Throughout this chapter, examples showing definitions of UNIX environment variables use Bourne shell syntax and refer to the Bourne shell startup files */etc/profile* and *.profile*. If you use the C shell, you should modify the examples to use proper C shell syntax and to refer to the equivalent C shell startup files */etc/cshrc* and *.cshrc*.

**Supported languages**

| Country or language | Country code | Code pages | Keyboard code |
|---------------------|--------------|------------|---------------|
| United States       | 001          | 437,850    | us            |
| Latin America       | 003          | 437,850    | la            |
| Netherlands         | 031          | 437,850    | nl            |
| Belgium             | 032          | 437,850    | be            |
| France              | 033          | 437,850    | fr            |
| Spain               | 034          | 437,850    | sp            |
| Italy               | 039          | 437,850    | it            |
| Switzerland         | 041          | 437,850    | sf,sg         |
| United Kingdom      | 044          | 437,850    | uk            |
| Sweden              | 046          | 437,850    | sv            |
| Germany             | 049          | 437,850    | gr            |
| Australia           | 061          | 437,850    | us            |
| Finland             | 358          | 437,850    | su            |

***NLS commands supported by DOS Services***

Some DOS NLS commands are supported only in VGA DOS sessions. You should use these commands only if you use one of these environments. The following table lists the DOS commands that are supported on each display type:

|                     | VGA displays <sup>3</sup>                    | ISO terminals        | Other displays       |
|---------------------|----------------------------------------------|----------------------|----------------------|
| CONFIG.SYS COMMANDS | COUNTRY <sup>4</sup><br>DEVICE= <sup>5</sup> | COUNTRY <sup>4</sup> | COUNTRY <sup>7</sup> |
| OTHER DOS COMMANDS  | CHCP<br>KEYB<br>MODE <sup>6</sup>            | CHCP                 | None                 |

3. In this table, "VGA Displays" are VGA DOS environments run inside an X window.
4. DOS Services does not support the optional third COUNTRY parameter, which specifies the COUNTRY.SYS file to use. DOS Services always uses \COUNTRY.SYS on the current drive.
5. DEVICE= entries for NLS features such as DISPLAY.SYS.
6. MODE options for NLS features such as CODEPAGE PREPARE.
7. On displays other than VGA or ISO, DOS Services supports only the first COUNTRY parameter, which specifies the country number.

### *DOS Services exceptions*

---

When you use DOS Services, you do *not* need to run the DOS NLSFUNC command as you would on a conventional DOS computer. DOS Services automatically runs NLSFUNC whenever you start DOS.

### *Sample NLS setup procedures for VGA displays*

---

Following are example NLS setup procedures for VGA displays. Both the startup file method and the DOS image method are described here. The instructions are based on a specific but typical example that assumes you want to type and display data in French instead of English. You can modify these instructions to match your own requirements by using different country codes, code pages, keyboard codes, and environment variable definitions.

Assume you want to use the DOS multilingual code page (850) and the French country code (33) by default.

### *Setting up the startup files*

---

Follow these procedures to set up your DOS NLS environment using the startup file method:

1. Add the following line to your home directory *CONFIG.SYS* file:

```
device=c:\usr\dbin\display.sys con=(ega,437,1)
```

This line sets up your VGA display so it can accept one code page in addition to the built-in hardware code page. The built-in hardware code page is 437.

**NOTE** Use "ega" in the "con=(ega,437,1)" portion of the "device=" line shown above for VGA displays.

2. Add the following line to your home directory *AUTOEXEC.BAT* file:

```
mode con codepage prepare=((850)c:\usr\dbin\ega.cpi)
```

This line prepares the VGA display for the additional code page (850), which is to be retrieved from the file named.

Note that you do not have to run the NLSFUNC command because DOS Services automatically loads NLSFUNC whenever you run DOS. Including an NLSFUNC command in *AUTOEXEC.BAT* does no harm, but it has no effect on DOS Services NLS functionality.



3. Define the UNIX CODEPAGE, COUNTRY, and KEYB environment variables by typing:

```
$ CODEPAGE=pc850
$ COUNTRY=33
$ KEYB=fr
$ export CODEPAGE COUNTRY KEYB
```

Add these lines to your home directory *.profile* file if you want them to be effective permanently.

Now, whenever you start DOS, code page 850 and country code 33 are set up automatically. You can also switch back to code page 437 by using the DOS CHCP command.

If you want code page 850 to apply when you run DOS applications directly from the UNIX prompt, be sure DOS Services interprets your home directory *AUTOEXEC.BAT* and *CONFIG.SYS* files. All DOS applications automatically interpret these files unless you have used the  $\pm p$  or  $\pm e$  options to change the default behavior. (See Chapters 5 and 6 for more information on DOS options.)

## Using DOS images

An alternative way to set up your DOS NLS environment is to create new DOS images loaded with the NLS commands that prepare the code pages you want to use. When you use the DOS image method, you need to make new images only when you want to use a code page that is not prepared in the current images. You must be logged in as root to perform the steps described in this section.

As with the previous example, we assume that you want to use the DOS multilingual code page (850) and the French country code (33) by default. Only the system administrator, logged in as root, can make DOS images, so you must be logged in as root to follow these procedures.

1. Modify the definitions of CODEPAGE, COUNTRY, and KEYB in */etc/profile* as follows:

```
CODEPAGE=pc850
COUNTRY=33
KEYB=fr
```

2. Change to the directory where the DOS VGA image is located by typing the following command:

```
cd /usr/merge/image/std/vga
```

3. Use a DOS editor to create two files with the same names as the DOS startup files: *autoexec.bat* and *config.sys*. If these files already exist, then use an editor to add the lines specified in the next step.
4. In the *config.sys* file, add:

```
device=display.sys con=(ega,437,1)
```

This line sets up your VGA display so it can accept one code page in addition to the hardware code page. The built-in hardware code page is 437.

**NOTE** Use "ega" in the "con=(ega,437,1)" portion of the "device=" line shown above for VGA displays. Also note that, unlike the previous example, the "display.sys" file on the "device=" line should *not* have a full path.

5. In the *autoexec.bat* file, add:

```
mode con codepage prepare=((850)ega.cpi)
```

This line prepares the display for the additional code page (850). Also note that, unlike the startup file method, the file *ega.cpi* should not have a full path.

6. Now make new images by typing the following command:

```
mkimg
```

Your DOS environment is now set up for NLS.

### *Resetting the DOS image*

---

If you want to remove the NLS commands from the image, follow these procedures:

1. Remove the lines you added to the *CONFIG.SYS* and *AUTOEXEC.BAT* files in the image directory. You may delete the files if they have no other commands in them.
2. Return the *CODEPAGE*, *COUNTRY*, and *KEYB* variables in */etc/profile* to their original values
3. Build new images by typing **mkimg**.

## Sample NLS setup procedures for ISO terminals

As before, this example assumes you want to use DOS code page 850 and country code 33. Note that some unique steps are required to set up NLS for use with an ISO terminal. These steps must be precisely followed.

1. Define the `ISO_TERM`, `CODEPAGE`, `COUNTRY`, and `KEYB` environment variables by typing:

```
$ ISO_TERM=1
$ CODEPAGE=pc850
$ COUNTRY=33
$ KEYB=us
$ export ISO_TERM CODEPAGE COUNTRY KEYB
```

Add these lines to your home directory *.profile* file if you want them to be effective permanently. The `ISO_TERM=1` coding tells the system that this is an ISO terminal. The `KEYB` environment variable must *always* be defined as "us".

2. Determine the current value of `LANG`. The current `LANG` value may be defined in either of two ways. First type the following command to see if `LANG` is defined as an environment variable:

```
$ echo $LANG
```

If this command displays a value (for example, "french\_france.ascii"), then this is the current `LANG` value.

If the `echo $LANG` command does *not* display a value, inspect the contents of the file `/etc/default/lang` by typing:

```
$ cat /etc/default/lang
```

This file contains a `LANG` definition in the form

```
LANG=language_territory.codeset
```

If the `echo $LANG` command does not display a value, your current `LANG` value is the value shown in `/etc/default/lang`.

For this example scenario, assume that the current `LANG` value is "french\_france.ascii".

3. If the code set portion of the `LANG` environment variable is *not* already "8859," redefine the `LANG` environment variable. To redefine `LANG`, type:

```
$ LANG=french_france.8859
$ export LANG
```

Add these lines to your home directory *.profile* file if you want them to be effective permanently. Note that ISO terminals *always* require the code set portion of the `LANG` definition to be 8859.

Now, whenever you run DOS, code page 850 and country code 33 are set up automatically. You can also switch to code page 437 by using the DOS CHCP command.

### ***Naming DOS and UNIX operating system files***

---

The DOS Services chapters of the User's Guide describe how DOS Services treats DOS and UNIX file names, including the DOS Services filename mapping feature. (DOS Services maps filenames when you use DOS to access a file created by the UNIX system with a name that is illegal in DOS. For example, DOS Services might map the UNIX filename *messageoall* to *MESS'BAQ* when you use DOS.) The description of file naming issues in the User's Guide applies in a multilanguage environment just as it does in a single-language environment.

In addition, the following points apply when you want to create or access files on the DOS Services system in a multilanguage environment:

- You can simplify file creation and file access by only using characters from "a" to "z" and "0" to "9" and the characters ".", "-", and "\_" in your filenames. All DOS code pages and UNIX code sets contain these characters.
- If you use the UNIX system to create filenames using characters that do not exist in your current DOS code page, DOS Services maps these file names to legal DOS names when you access them with DOS. For example, the UNIX filename *ms-dos®* might be mapped to *MS-D'BDQ* when you use DOS. DOS Services maps this filename because the character "®" exists in some UNIX code sets but does not exist in supported DOS code pages. Refer to The DOS Services chapters of the User's Guide for further information on using mapped file names.
- You *cannot* use DOS to create filenames in the shared filesystem using characters that do not exist in the UNIX code set. You *can* create these files in real DOS filesystems, such as the DOS partition or a diskette drive. For example, if the character "μ" does not exist in the UNIX code set, you cannot use DOS to create a file named *μNET* on drive C: no matter what DOS code page you use. If your current DOS code page does include the "μ", you can use DOS to create *μNET* on drive E:, drive A:, or any other real DOS filesystem.

## Printing in an NLS environment

By default, DOS Services sends DOS printer output to the UNIX print spooler.<sup>8</sup> DOS Services by default does *not* translate DOS text when you print. If the UNIX printer used for DOS printing does not support all the characters you send for printing, the results are unpredictable.

You *cannot* use the DOS **MODE** or **CHCP** commands to change the printer code page when you use a UNIX printer. However, when you use the UNIX print spooler, you can use the DOS Services **charconv** command to translate DOS text files before you print them. For example, assume your DOS text file **MEMO** was created using DOS code page 850 and you want to print it using the UNIX ISO standard code set (8859) and convert characters that do not exist in code set 8859 to the best multibyte approximation. You could convert **MEMO** using the command:

```
C: charconv /i pc850 /o 8859 /m /d memo memo.unx
```

You can then print the converted file using standard DOS or UNIX commands (for example, **copy memo.unx lpt1**).

You can also use the DOS Services **printer** command to change the default UNIX print command that DOS Services uses when you send DOS printer output to the UNIX spooler. For example, if you want to use the **charconv** command shown above whenever you print from DOS, you could issue the following command at your DOS prompt or in your **AUTOEXEC.BAT** file:

```
C: printer unix "charconv -i pc850 -o 8859 -m -d | lp"
```

After you issue this **printer** command, all text files you print from DOS are converted automatically.

Refer to the next section, "Converting text files," for more information on the **charconv** command. See Chapter 2 for more information on the **printer** command.

Instead of using the UNIX print spooler, you can attach a printer directly to your DOS process. When a printer is directly attached to DOS, you can use the DOS **MODE** or **CHCP** commands to change printer code pages and use it in all other ways exactly as you would on a conventional DOS computer. See Chapter 2 for further information on setting up and using a directly attached printer.

8. DOS Services uses the script `/usr/lib/merge/lp` to spool DOS printer output.

**NOTE** The examples using **charconv** and **printer unix** in this section are relevant only for text files. It is not necessary to convert DOS graphics or other nontext files before printing.

### *Converting text files*

---

DOS Services has three commands that you can use to convert text files so they are usable with different code pages and code sets. The commands are **dos2unix**, **unix2dos**, and **charconv**. All of these commands are usable both in the DOS environment and at the UNIX shell prompt.

### *Using dos2unix and unix2dos*

---

The DOS Services chapters of the User's Guide describes how to use **dos2unix** and **unix2dos** to convert text in DOS format to UNIX format and vice versa. In addition, **dos2unix** and **unix2dos** by default translate each character in your text from the DOS code page to the corresponding character in the UNIX code set or vice versa. When you use these commands in the DOS environment, DOS Services translates between your current DOS code page and the UNIX code set defined by the **LANG** environment variable. When you use them at the UNIX shell, DOS Services translates between the DOS code page defined by the **CODEPAGE** environment variable and the UNIX code set.

When all characters in your text file exist in both the code page and the code set, the converted output looks exactly the same as your original, unconverted text file (except that it is now usable with a different operating system). When your original text file contains characters that do not exist in both the code page and code set, **dos2unix** and **unix2dos** by default convert untranslatable characters into asterisks (\*).

You can modify the behavior of **dos2unix** and **unix2dos** by using the same options that apply to the **charconv** command, which is described in the next section.

### *Using charconv*

---

The **charconv** command is a flexible and powerful text conversion tool. **charconv** is most useful in mixed-language (NLS) environments. For simple file conversion cases, use **dos2unix** and **unix2dos**. Unlike **dos2unix** or **unix2dos**, **charconv** has no defaults for code pages and code sets. The **/I** and **/O** options are required unless you use the **/X** option.

The syntax of the **charconv** command is:

```
charconv [options] sourcefile [targetfile]
```

The source and target files must not be the same. When neither file parameter is specified, **charconv** prints a usage message to your screen. If you specify only one file parameter, that file is considered the source file, and the output is written to standard output.

### ***charconv options***

When issued from the UNIX command line, **charconv** options must be specified with a hyphen (-) rather than a slash (/). In the DOS environment, you can use either the slash or the hyphen. The examples here show the hyphen, which will work in either environment. You can use either uppercase or lowercase options both from the UNIX shell and in the DOS environment.

All of the options below can be set on the command line or with the **CONVOPTS** ("charconv options") environment variable, which is described later.

- /?** Displays a list of valid **charconv** options on your screen.
- /7** Notifies you if **charconv** encounters any 8-bit characters and translates them to 7-bit characters. You cannot use this option with **/B**.
- /A** Causes **charconv** to quit if it encounters a character it cannot translate accurately. This option overrides the **/C**, **/M**, and **/S** options, which allow inexact translations.
- /B** Preserves 8-bit (binary) character representations. This option is on by default.
- /C x** Specifies *x* as the character to use for untranslatable characters. The default character is an asterisk (\*).
- /D** Converts a file from DOS to UNIX format by removing carriage returns from the end of each line.
- /I tbl** Identifies the code page or code set used for the input file. This option is required unless you use the **/X** option. Refer to the Release Notes for a list of supported code pages and code sets.

#### ***/O tbl***

Identifies the code page or code set used for the output file. This option is required unless you use the **/X** option. Refer to the Release Notes for a list of supported code pages and code sets.

For example, if you wanted to convert a DOS file from code page 437 to code page 850, use the **charconv** command as follows:

```
charconv -i pc437 -o pc850 sourcefile targetfile
```

- /L** Converts text to lowercase.
- /M** Converts single, untranslatable characters into multibyte characters, if possible. For example, ¾ is converted to 3/4. If a multibyte conversion is not possible, **charconv** uses the default asterisk (\*) character.

- /P** Converts a text file from UNIX to DOS format by adding carriage returns to the end of each line.
- /Q** "Quiet." Prevents **charconv** from printing warning messages and character translation statistics to your screen. This is the default. Use **/Q** if necessary to override the **-V** option, which causes warning messages and conversion statistics to be displayed.
- /S** Converts single, untranslatable characters into the best single character translation. For example, **\*I** (the DOS graphic character for intersection) is converted to **+**. If there is no best single translation, **charconv** uses the default asterisk (**\***) character.
- /U** Converts text to uppercase.
- /V** Displays warning messages and character conversion statistics on your screen.
- /X** Specifies that **charconv** should not translate from code page to code set or vice versa. Use this command with the **/D** or **/P** options when you want to translate between the DOS and UNIX operating systems without changing code pages.  
  
When you specify **/X**, **charconv** does not allow you to use the **/I** or **/O** options.
- /Z** Causes **charconv** to stop processing when it encounters the DOS end-of-file (EOF) character, **<Ctrl>Z**. This option removes the end-of-file marker when converting text from DOS to UNIX. By default, **charconv** converts the whole file.

### *Using the CONVOPTS environment variable*

The **CONVOPTS** environment variable allows you to set **charconv** options once so you only need to type **charconv filename** to accomplish a task.

For example, suppose you want to convert a series of files from code page 437 to code page 850 and make all of the characters uppercase.

If you use **charconv** in the DOS environment, set the **CONVOPTS** environment variable on the DOS command line as follows:

```
set convopts=/u /i pc437 /o pc850
```

If you use **charconv** at the UNIX shell, define the **CONVOPTS** variable as follows (this example assumes you use the Bourne shell):

```
CONVOPTS="-u -ipc437 -opc850"
export CONVOPTS
```

Now, you can just type the following line for each file:

```
charconv sourcefile targetfile
```

and **charconv** automatically converts from code page 437 to 850 and makes all characters uppercase.



## Chapter 8

# Other advanced topics

---

This chapter provides information of interest both to advanced DOS Services users and to the DOS Services system administrator. It covers a wide variety of techniques for customizing DOS and improving DOS Services efficiency. It is organized as follows:

- “Changing Colors in DOS Windows” describes the mechanisms offered by the SCO Open Desktop/SCO Open Server and by DOS Services for adjusting colors.
- “X System Resources for DOS Services” describes the X resources users can set to control the appearance of their DOS windows.
- “Using Expanded Memory (EMS)” describes the *CONFIG.SYS* entry for the DOS Services EMS device driver.
- “Using Multiport Serial Cards” provides hints for using these cards with DOS processes.
- “DOS Services Compatibility” compares the SCO system environment with a conventional, stand-alone DOS environment and discusses the requirements that DOS applications must observe in order to operate correctly with DOS Services.
- “Passing UNIX Environment Variables to DOS” shows you how to customize several DOS features by using UNIX environment variables.
- “Running DOS or a DOS Application as the Default Shell” tells you how to start DOS or a specific DOS application as your default environment when you log into your DOS Services computer.
- “Scheduling DOS Programs” illustrates how you can use the UNIX *at*, *batch*, *cron* and *nohup* commands to control the execution of DOS processes.

## *Changing colors in DOS windows*

---

The X server you run may affect the color response your DOS programs display when you run them in a window. Systems running a 256-color X server will receive the same color response in an X window that the DOS program would give on a standard DOS system. Systems running a 16-color X server may require some extra tuning.

When you run DOS Services programs in an X window on a 16-color server, you may occasionally need to redefine the colors appearing on your screen. To do this correctly, you need to understand the layers of software that are interacting and how they handle colors. The following discussion applies only to 16-color X server systems.

### *How DOS and X Windows handle colors*

---

A 16-color X server places a ceiling on the number of colors you can use for each window on your screen.

SCO provides a special X Client, **Color**, which allows each user to choose his own Color Palette, a set of eight shades (plus black and white). Once you choose such a palette, it must serve the needs of all the X clients you run, including any DOS program you run in an X window.

When running in full-screen mode, DOS Services supports all the normal DOS video standards. You can use MDA, Hercules, CGA, EGA or VGA. When running a DOS program in an X Window, however, EGA and VGA display modes are not supported. On a 16-color X server, text screens can display up to 16 colors on the screen at one time. These are predefined colors which cannot be changed. Graphics screens can be 4-color CGA or Hercules graphics.

When you zoom the window, the program running in that window gains full control of the screen. A DOS program running in a zoomed window will give you exactly the same color response it would under ordinary DOS. If you are running a VGA program, with a VGA monitor and with DOS Services set for VGA mode, the program has access to the full range of VGA colors, up to 256 in some VGA modes. Other VGA modes offer 16 colors, but the user can select from an array of 64 or 256 hues.

When you are running under X Windows, however, the X server limits the number of colors available to each window. When your screen contains both UNIX and DOS windows, color scheme conflicts can occur because DOS programs and the SCO system have different color requirements.

Motif Windows achieve their sculptured appearance through shadowing effects, which require many shades of the same basic color. DOS screens are often designed around the assumption that they have access to a fixed array of contrasting primary colors. Consequently, when the number of colors available to your screen is small, no single set of tints will adequately serve both purposes. The range of subtle shades suitable to Motif window borders and dialog boxes can produce DOS screens in which the text is unreadable. The contrasting primary colors used by DOS programs generate Motif screens with a bizarre technicolor appearance.

Consequently, when you run both DOS and UNIX programs in windows and you move back and forth between them, you sometimes need to change your color selection.

## *Changing the color scheme of your windows*

For the purpose of the following discussion, some special terms are defined:

- **Color** - an X client that allows you to redefine the colors appearing on your screen.
- **Color Palette** - a group of eight colors, defined by the **Color** utility and serving the needs of all your X clients.
- **UNIX Color Palette** - a group of eight colors, oriented toward serving the needs of your UNIX X clients, which run in Motif windows.
- **DOS Primary Colors** - the name given to a particular palette of eight primary colors. These colors were chosen as those most closely simulating the standard DOS sixteen-color environment.
- **DOS Colors** - the sixteen standard DOS primary colors.

When your SCO system session starts, it comes up using a default Color Palette. If you want to be able to change colors during your session, start the **Color X Client** by clicking its icon. This provides a selection of predefined palettes. For full details on the use of the **Color** utility see the User's Guide.

Among the palettes offered to you by **Color** is one marked **DOS Services**. This is a selection of the eight primary colors most often used by DOS programs. When you make this selection, most DOS programs will present readable text. Your UNIX screens may or may not be acceptable. This is likely to provide a workable compromise for those situations where you will be alternating between DOS and UNIX activities.

The compromise is not perfect, however. It still gives you only 10 colors (eight primaries plus black and white). The eight colors have been chosen as those most closely approaching the needs of DOS's sixteen standard colors, but there are still some legitimate DOS combinations which will be

unreadable. There is no specific match for the color DOS knows as "grey," for instance. Any program displaying grey text on a background of the color to which grey is mapped would be unreadable.

You can handle the situation in several ways:

1. Reconfigure the DOS program. Many DOS programs allow you to select the colors you wish to use. Simply pick color combinations that lie within the scope of the "DOS Services" palette.
2. Zoom the Window. DOS programs operating in a zoomed window have access to whatever colors your monitor offers.
3. Use the DOS Colors option. The DOS Menu offers a button marked DOS Colors. This installs the sixteen official DOS colors. The procedure is described below.

### *Installing the DOS colors*

---

To install the sixteen standard DOS colors as the color group being used for all your X windows:

1. In any window running DOS Services, bring up the DOS Menu by holding down the right mouse button or typing `(Alt)D`.
2. From the DOS Menu, select DOS Colors.

These colors have not been optimized for UNIX work. Therefore your DOS windows will look correct, but any UNIX windows that may peek out from behind a DOS window will display wildly distorted shading. Since you usually work in only one window at a time, this has no effect on your productivity. When you switch to one of your UNIX windows, the last color palette selected through Color is automatically installed.

### *X System resources for DOS Services*

---

Users can control the appearance of their DOS window through a standard set of X resources. There are many X Resources available. This section lists only those that are really useful. Users can define them in their own personal resource file, `.Xdefaults`. The default values can be found in `/usr/lib/X11/app-defaults/DOS`. You can assign values to any of these resources using the standard X syntax:

**`dos*resource: string`**

- **autofreeze** - When set to yes, DOS applications will be suspended in windows when the keyboard focus has been moved to another window. The default is no.

- **autozoom** - When set to yes, applications will zoom automatically when they go into VGA graphics mode under X Windows. The default is no.
- **colors** - Sets the colors to be used by the driver.
- **dos\_colors** - The normal method of selecting colors in DOS windows is through the **Color** utility or the "DOS Colors" option in the DOS Menu. We recommend that users apply these methods for all their ordinary needs. Those few users who have very special needs for precise color matching between their DOS and UNIX windows can adjust this mapping by manipulating the **dos\_colors** variables.

When the **installcolormap** resource is set to false, DOS Services uses the default UNIX colormap and attempts to find UNIX colors that best match the needed DOS colors. Users can try to fine-tune this color matching by specifying another UNIX color to be matched to a particular DOS color. For example, the following statement tells DOS Services to look for the UNIX color "navyblue" to use as the DOS color "blue":

**dos\*blue: navyblue**

If "navyblue" is not found, DOS Services uses its own built-in match.

Following is the list of DOS colors that can be set:

|           |              |
|-----------|--------------|
| black     | gray         |
| blue      | lightblue    |
| green     | lightgreen   |
| cyan      | lightcyan    |
| red       | lightred     |
| magenta   | lightmagenta |
| brown     | yellow       |
| lightgray | white        |

See the DOS Services system-wide resource file `/usr/lib/X11/app-defaults/DOS` for the current default settings.

- **dosfont** - Sets the default font. DOS Services supplies the following PC fonts:

6x13pc  
8x14pc

**NOTE** Any fixed font can be used as the default font. However, if you plan to take advantage of the National Language Support (NLS) facilities provided under DOS Services, then you should only use the fonts shown above.

- **installcolormap** - By default, the DOS window uses the color palette established by the **color** utility (**installcolormap: false**). This is sufficient for most purposes. When standard DOS colors are required in the DOS

window, the user can set this variable to **installcolormap: true**. This installs the standard DOS colors for every session. The default is false.

Use of this variable is not encouraged. The DOS Colors button on the DOS Menu installs the standard DOS colors temporarily, for the period of that DOS session only. Since need for true DOS colors is usually infrequent, this method is preferred.

- **menukey** - Specifies the key sequence for invoking the DOS menu. The default is **(Alt)D**. The key sequence can be set to any combination of the **(Ctrl)**, **(Alt)**, and **(Shift)** keys and an ASCII character. Use the standard X name for the key, delimited by **<** and **>**. For example, to set the key sequence to **(Ctrl)(Shift)(F10)** key, you would set the following in your resource file:

```
dos*menukey:ctrl shift (F10)
```

The DOS menu can also be invoked by pressing and holding the right mouse button.

- **windowsHeight** - Sets the height of X windows while in Microsoft Windows mode. The default is 80% of your screen height.
- **windowsWidth** - Sets the width of X windows while in Microsoft Windows mode. The default is 80% of your screen width.

## *Running DOS Services on remote servers*

---

### *Setting the XMERGE environment variable*

In order for DOS Services to display properly on a remote server, the **XMERGE** environment variable must be set on that server before the DOS process starts. It is recommended that it be set to **cga** mode. When DOS Services is run locally, this environment variable is set automatically.

To set the **XMERGE** variable, use the following commands:

Bourne or Korn shell

```
XMERGE=cga; export XMERGE
```

C-shell

```
setenv XMERGE cga
```

### *DOS Services fonts*

When DOS Services is run remotely, all font files it uses must be installed and compiled on the remote server's machine. To determine the names of the font files to install, examine the file **/usr/lib/merge/fonts/fonts.src**. This file is in two-column format, with the names of all of the fonts in the first column and the

corresponding filenames in the second column. The actual font files are obtained from the subdirectory *misc* in the server's standard font directory.

## Using expanded memory (EMS)

The DOS section of the User's Guide describes expanded memory from the DOS Services user's perspective. Advanced users or the system administrator may need to understand the EMS device configuration command in the *root* directory *CONFIG.SYS* file. Because this command is in the system default *CONFIG.SYS* file, it is interpreted whenever any user starts a DOS environment or a DOS application, unless the user specifically requests that *CONFIG.SYS* not be interpreted. The *CONFIG.SYS* command is:

```
device=c:\usr\lib\merge\emm.sys
```

Due to built-in defaults in the driver, the above line is equivalent to the following:

```
device=c:\usr\lib\merge\emm.sys /start=D000 /address=208
```

In this line, "208" is the base port number used for expanded memory, and "D000" is the segment address in DOS space to which expanded memory is mapped. Do not use a value other than 208 for the base port number. However, you can use a segment address other than D000 if you have other devices that use that address. If you cannot use D000 because of a conflict like this, change the EMS device configuration command in *CONFIG.SYS* to use one of the following segment addresses instead:

|   |      |      |      |      |
|---|------|------|------|------|
| 1 | C000 | C400 | C800 | CC00 |
| 1 | D400 | D800 | DC00 | E000 |

## Using multiport serial cards

Many cards provide multiple serial ports on a single adapter. Each of these is a separate UNIX device, and UNIX processes can direct output to any of them. DOS, however, recognizes only a limited number of serial ports.

Standard practice assigns UNIX devices */dev/tty1a* and */dev/tty2a* to COM1 and COM2 respectively, but DOS Services allows you to assign any of the available UNIX devices.<sup>1</sup> To use a serial port other than the standard devices for DOS COM1 or COM2, issue the *dos* command with the *+a* option. Attach COM1 or COM2 to your DOS process and map COM1 or COM2 to the actual UNIX name

1. DOS Services does not support "Smart" Multiport Serial cards unless the manufacturer of these cards has added special support.

of the serial port. For example, to run a DOS environment and attach `/dev/tty20` as COM1, use the command:

```
$ dos +acom1=/dev/tty20
```

You must use indirect attachment, as illustrated in this example, to attach a serial port other than COM1 or COM2. Attempts to use direct attachment, as shown in the following example, do not work:

```
$ dos +adcom2=/dev/tty15 (incorrect)
```

Refer to Chapter 6 and to the DOS section of the User's Guide for further information on using COM ports.

## *DOS Services/DOS compatibility*

---

The following paragraphs summarize the significant technical differences between DOS as it runs on a conventional, stand-alone personal computer and DOS as it runs under DOS Services. Despite these differences, nearly all DOS applications that run correctly under DOS also run correctly under DOS Services.

### *DOS interrupts*

---

DOS commands and applications that use standard DOS interrupt 21 function calls work properly with DOS Services. Commands and applications that attempt to use interrupts 13, 25, or 26 to access the shared filesystem do not work.

Commands that use interrupts 13, 25, or 26 include **CHKDSK**, **FDISK**, **FORMAT**, and **SYS**. These commands are described as "exceptional" because they do not work on the filesystem shared by DOS and the UNIX operating system.

A few applications use interrupts 13, 25, or 26 to perform low-level disk I/O operations such as:

- "Unerasing" deleted files.
- Repacking segments on the fixed disk to reduce fragmentation.
- Changing segment interleaving on the fixed disk.

You cannot use these applications to modify the filesystem shared by DOS and the UNIX system.

You can use the commands and applications described above on a real DOS filesystem, such as a diskette drive, a physical or virtual DOS partition, or a virtual floppy.

If you try to use any of these commands or applications to modify the shared filesystem, DOS displays an error message and the operation fails, but your data and programs are not harmed in any way.



## *DOS clock interrupt rate*

---

By default, DOS Services uses the standard DOS clock interrupt rate of approximately 18 clock ticks per second. All DOS applications that work at this clock interrupt rate should work in the DOS Services environment. DOS Services has an option that can change the clock interrupt rate to one per second, which improves the performance of many applications. See the description of the **merge** command **set fastclk** option in Chapter 4. The slower clock interrupt rate is compatible with most, but not all, DOS applications.

## *Applications that continuously poll the keyboard*

---

Unlike standard DOS, DOS Services automatically puts most DOS applications that continuously poll the keyboard to sleep until there is keyboard input. DOS Services has an option that can prevent this default behavior. See the description of the **merge** command **set pollsleep** option in Chapter 4.

## *Configuring applications that buffer keystrokes*

---

Some DOS applications that buffer keystrokes may not work as expected in the DOS Services environment. These applications include applications, such as SuperKey, that create keyboard macros by mapping multiple keystrokes to a single key.

You can correct this problem by putting the following line in a **CONFIG.SYS** file that is interpreted when you run DOS:

```
device=\usr\dbin\ansi.sys
```

This approach has a side effect, however. To improve system response, DOS Services, by default, puts DOS applications that poll the keyboard to sleep while they wait for keyboard input. When you include the **device=\usr\dbin\ansi.sys** line in your **CONFIG.SYS** file, DOS Services no longer puts applications that poll the keyboard to sleep. Your computer is therefore likely to be more heavily loaded, especially if you run multiple DOS applications at once.

## *Local and remote drives and file handles*

---

Unlike standard DOS, DOS Services treats drives that access the filesystem shared by DOS and the UNIX system (C:, D:, and J:) and any files contained in them as remote drives and remote files in a network. DOS Services has options that cause DOS to interpret these drives and files as local. See the description of the **merge** command **set drive** and **set handle** options in Chapter 4.

## *Limits on number of open files*

---

DOS applications may not open more than 123 files in the shared filesystem using the new-style open calls (that is, require the presence in the CONFIG.SYS file of a FILES=*n* command, where *n* is greater than 123).

## *Protected mode restrictions*

---

DOS Services allows you to access the High Memory Area. In this sense, you manage DOS memory as you would on an AT-compatible computer. In all other important respects, however, the DOS Services environment emulates a PC-compatible computer.

DOS Services creates a DOS environment by invoking the *Virtual 8086* mode of the Intel 80386 or 80486 processor. Any application you run in such an environment must be compatible with the Intel 8086 processor. Any DOS application absolutely requiring an 80286 or later processor does not run under DOS Services, except for Windows 3.1 and Windows 3.1 applications.

Despite the fact that the application is running on an 80386 processor, as a general rule, it cannot make use of CPU capabilities beyond those provided by the 8086. In particular, it cannot access the processor's Protected Mode for multitasking or its ability to address memory beyond the one-Mbyte DOS limit. Any DOS application attempting to do so will misbehave in some way. It may exit gracefully, hang the application or kill your DOS session.

Most DOS applications are meant for the 8086. Of those designed for more powerful processors, many are smart enough to sense the nature of the processor on which they are running and adjust their operation accordingly. For these "intelligent applications" you need to do nothing special. They will respond to the V86 environment as an 8086 processor and automatically run in the 8086-compatible "real mode".

For applications that offer 286/386 features and fail to adjust to the V86 automatically, there are several ways to compensate. See Chapter 3 for details.

In making your choices as to which DOS software to run under DOS Services, consider the following factors:

- **The amount of memory used by the program is not the problem; it is the way that memory is accessed.** Programs that use Expanded Memory will work; programs that use Extended Memory will not. DOS Services fully supports DOS-based Expanded Memory, using the LIM 4.0 interface. DOS programs that use Expanded Memory will work properly under DOS Services. Extended Memory, however, is available only in the processor's 286/386 mode. Programs that use Extended Memory will not run under DOS Services. (DOS Services allows a certain limited access to the High Memory Area, but all other Extended Memory is unavailable.)

- **Multitasking per se is not restricted; the issue is 286/386 features.** Any application which successfully performs multitasking on a standard 8086 DOS computer should run properly under DOS Services. Any application which uses the processor's Protected Mode for its multitasking will not.
- **Except for Windows 3.1, if the program does not run on a PC or XT, it will not run under DOS Services.** Any program requiring an 80286 or higher AT-style processor cannot be used under DOS Services. Consult the program documentation under "System Requirements."
- **You may not require the application at all.** Most applications using 286/386 features are aimed at multitasking or managing large amounts of memory. SCO UNIX is a full-fledged, multiuser, multitasking operating system, allowing each user to run many programs at once. With DOS Services, this can include any combination of UNIX or DOS applications. The UNIX operating system can easily run very large applications. Many DOS programs have been ported and your application may be available for the UNIX operating system.

## Passing UNIX environment variables to DOS

---

DOS and the UNIX system use similar concepts of environment variables (also called "environment strings"). An environment variable is a value (assigned by you, the operating system, or an application) that is available to all commands and applications that run in the same environment. Refer to your DOS documentation for further information on environment variables.

You can set DOS environment variables with the DOS `set` command. By default DOS Services sets five DOS environment variables when you start DOS. To view their current values, type `set` at the DOS prompt. The output is similar to the following:

```
COMSPEC=C:\usr\dbin\command.com
PATH=C:\BIN;C:\USR\BIN;C:\USR\DBIN;C:\USR\LDBIN;
LANG=En
TEMP=C:\TMP\
TMP=C:\TMP\
```

Values are assigned to these variables as follows:

- **COMSPEC** is set to the `COMMAND.COM` file that is used with DOS Services.
- **PATH** is set to match the UNIX path. Note that the UNIX path includes the standard directories for DOS programs, `/usr/dbin` and `/usr/ldbin`.
- **LANG** is set to be the same as the UNIX `LANG` environment variable.
- **TEMP** and **TMP** are set to `C:\TMP\`, which is the preferred temporary directory.

DOS applications may assume that you explicitly assign the values of required environment variables yourself. The Microsoft C Compiler, for example, assumes that you have assigned a value to the environment variable called **INCLUDE**, which tells the compiler where to look for *#include* files if they are not found in the current directory. To assign the value "**\USR\LDBIN\INCLUDE**" to the environment variable **INCLUDE** on DOS drive C:, type:

```
C: set include=c:\usr\ldbin\include
```

DOS environment variables are typically set in the *AUTOEXEC.BAT* file, so they are available for use by any command or application whenever you use DOS.

You can use DOS environment variables in the DOS Services environment exactly as you would use them on a conventional DOS computer. DOS Services, in addition, has special features for manipulating DOS environment variables that:

- Can override the default DOS search path automatically.
- Set other DOS environment variables so they are available to DOS programs run either in the DOS environment or from the UNIX shell.

These features improve DOS performance and flexibility.

### *Setting your DOS search path with DOSPATH*

---

Because DOS Services automatically sets the search path used for DOS programs, it is rarely necessary for you to change it. You can change your UNIX search path at any time using the procedures described in your UNIX manuals.

You can, if necessary, reset the DOS search path in the DOS environment so it is different from your UNIX search path. You can reset your search path manually by using the DOS **PATH** command at the DOS prompt, or else override the default path in one of two ways:

1. Set your DOS search path with the **PATH** command in your home directory *AUTOEXEC.BAT* file.
2. Use the UNIX environment variable **DOSPATH** to communicate your preferred search path to DOS when you enter the DOS environment.

The second method (using **DOSPATH**) is preferred because:

- It is more efficient than running *AUTOEXEC.BAT*.
- Since *AUTOEXEC.BAT* is not always executed when you start a DOS environment or run DOS programs from the UNIX shell, any search path specified in *AUTOEXEC.BAT* may not be effective from the UNIX shell. The **DOSPATH** value, on the other hand, is always used.

To set your preferred DOS search path using the **DOSPATH** environment variable, follow these steps:<sup>2</sup>

1. At the UNIX system prompt, type **DOSPATH=** followed by the desired DOS search path. You can use either the UNIX slash (/) or the DOS backslash (\) as the path separator. If the path includes backslashes or multiple directories separated by semicolons, enclose the path in single quotes (') to prevent the UNIX shell from interpreting the backslashes or semicolons. For example:

```
$ DOSPATH='c:/usr/dbin;c:/usr/ldbin;c:/usr/mary/dbin;a'
```

2. Export **DOSPATH** as you would any UNIX environment variable, by typing:

```
$ export DOSPATH
```

You can use any valid DOS drive designator (such as a:, c:, d:, e:, or j:) in the **DOSPATH** definition. All directories in the shared filesystem that you include in your path definition must be specified using valid UNIX names, including correct use of upper- and lowercase letters. Use UNIX names rather than DOS mapped names for any directory names that are not legal in DOS. Invalid UNIX names are not included in the DOS search path.

You can include the **DOSPATH** definition and **export** instruction in your home-directory *.profile* file if you want your specified DOS path to take effect every time you log into the DOS Services system.

The system administrator can include a default **DOSPATH** definition in */etc/profile* that affects all users. Users can override a system **DOSPATH** with their own **DOSPATH** definition or append additional paths using a command in the form:

```
$ DOSPATH=$DOSPATH'newpath'
$ export DOSPATH
```

where *newpath* is the list of additional directories the user wants searched.

## Setting other DOS environment variables with DOSENV

Use the UNIX **DOSENV** environment variable to set any DOS environment variables except for **PATH** or **COMSPEC** from the UNIX shell.

2. The examples in this section describe setting variables in the UNIX system Bourne or Korn shell. If you are working in the C shell, use the **setenv** command and do not export the variable.

To use DOSENV, follow these steps:

1. Set the required DOS environment variables using standard UNIX syntax:

```
$ VARIABLE1=VALUE1
```

```
$ VARIABLE2=VALUE2
```

```
.
. .
.
```

2. Assign to the UNIX environment variable DOSENV the names of all the DOS environment variables that you defined in step 1. Use commas (but no spaces) to separate the variable names:

```
$ DOSENV=VARIABLE1,VARIABLE2,...
```

3. Export each of the defined DOS environment variables and DOSENV:

```
$ export VARIABLE1 VARIABLE2... DOSENV
```

Assume, for example, that you want to:

- Change your DOS prompt to show your current working drive and directory by using the DOS PROMPT environment variable.
- Assign the value C:\USR\LDBIN\INCLUDE to the environment variable INCLUDE, used by the Microsoft C Compiler.

To accomplish these operations, you would type:

```
$ PROMPT='pg'
```

```
$ INCLUDE='c:\usr\ldbin\include'
```

```
$ DOSENV=PROMPT,INCLUDE
```

```
$ export PROMPT INCLUDE DOSENV
```

Observe the following concerning this example:

- Quotes are used around \$p\$g in the first line to prevent the UNIX shell from interpreting the metacharacter "\$". (For further information on the syntax for the PROMPT environment variable, refer to your DOS documentation on the PROMPT command.)
- In the second line, DOS-style backslashes are used instead of UNIX-style slashes because DOSENV passes the values of all variables to DOS literally, with no translation. The string c:\usr\ldbin\include is surrounded by single quotes to prevent the UNIX shell from interpreting the metacharacter "\".
- Items in the list of environment variables following DOSENV= are separated by commas, with no spaces.

You can define DOS environment variables using DOSENV in your home-directory .profile file to make them effective every time you log into DOS Services. The system administrator can include DOSENV definitions in /etc/profile that affect all users. Users can override a system DOSENV with their own

DOENV definition or append additional DOENV variables to the system definition with a command in the form:

```
$ DOENV=$DOENV,VARIABLE1,VARIABLE2,...
```

where *VARIABLE1*, *VARIABLE2*, and so on, are the user's own preferred DOENV definitions.

## *Running DOS or a DOS application as the default shell*

---

When configured to support DOS Services, your computer runs the standard UNIX System V operating system. The Bourne shell is the default shell when you log in. When you type:

```
$ dos
```

DOS Services presents you with a continuous DOS environment — an alternative shell that understands DOS commands.

You can use the DOS shell or a specific DOS application immediately when you log in, without typing *dos* or the application name, by invoking DOS or the application in your home directory *.profile* file.

### *Running DOS as the default shell*

---

To run DOS as the default shell, add the *dos* command to the end of your *.profile*, optionally followed by the *exit* command.

The *dos* command starts a DOS environment automatically as soon as you log in. You leave the DOS environment in the normal way — by typing *quit*. If the *dos* command is not followed with an *exit* command in your *.profile*, you return to the UNIX shell prompt when you type *quit*. You can then continue using the UNIX system or log out in the usual way. If the *exit* command follows the *dos* command in *.profile*, typing *quit* terminates your DOS environment and logs you out of the UNIX system.

### *Running a DOS application as the default shell*

---

If you want a specific DOS application to start automatically when you log into your DOS Services computer, follow procedures similar to those described in the previous section. For example, if you want to run Lotus 1-2-3 immediately upon login every time you use your computer, add the *123* command (instead of *dos*) to the end of your home directory *.profile*, optionally followed by the *exit* command. Then when you log in, Lotus 1-2-3 starts automatically. When you exit Lotus, you either return to the UNIX shell or are logged out, depending on whether you include the *exit* command in *.profile*.

## Scheduling DOS programs

---

You can use the UNIX commands **at**, **batch**, **cron** and **nohup** with DOS processes just as you use them for UNIX processes. The **at** command runs DOS or UNIX commands at a later time that you specify. The **batch** command works similarly but executes your specified processes when the system load allows rather than at a time you specify. The **cron** command is also similar but is used for commands that are to be run regularly at a specified time. The **nohup** command starts a process that is immune to hangups and quits.

You can run any of these commands from the UNIX shell using standard UNIX syntax and including any valid options in the DOS command. For example, you could tell the Microsoft C Compiler to compile and link the file *BIG.C* at 8:00 P.M. by typing:

```
$ at 2000
cl big.c
(Ctrl)D
```

You do not need to be logged in at the time the command executes, as long as the system is running. For further information on these UNIX scheduling commands, consult your UNIX documentation.

When you use these scheduling commands with DOS programs, observe the following precautions:

- You cannot use these scheduling commands with display-oriented DOS programs. (See the description of the **±b** option in Chapter 6 for more on display- and stream-oriented programs.)
- Output is treated in the standard UNIX way for the scheduling command you use. If you do not redirect the output, **at**, **batch** and **cron** mail the output to you. The **nohup** command causes output to be directed to the file **nohup.out** unless you specify otherwise.
- If the DOS program requires a key disk, the key disk must be in the diskette drive at the time the program runs.
- You cannot send input from your keyboard to programs started with these scheduling commands. If the program accepts redirected input, you can redirect keyboard input from a file at the time you issue the command. For example:

```
$ at 2000
doit < kbdinput > output
(Ctrl)D
```



## *Appendix A*

# *Using the system console and terminals*

---

DOS Services works from a wide variety of displays, including:

- The UNIX system console, which can use any of several different types of display adapters.
- PC scancode terminals, which emulate some of the features of the console.
- Serial terminals, including those that display only ASCII characters and those that are enhanced to display some or all of the full IBM PC character set used by DOS.

DOS Services characteristics differ depending on the capabilities of your display. This appendix describes the features and limitations of DOS Services when you use each type of display.

## *Characteristics of DOS on the console*

---

DOS Services works on any UNIX system console that consists of a monochrome or color monitor and a PC-compatible keyboard connected to one of the following display adapter boards:

- a Video Graphics Array (VGA) adapter
- a Color Graphics Adapter (CGA)
- a Hercules adapter
- a Monochrome Display Adapter (MDA)

In CGA, Hercules, EGA, and VGA modes, the console can display bit-mapped graphics. In CGA, EGA, and VGA modes, the console can display color.

Just as on a conventional DOS computer, you cannot run DOS applications that require a more powerful display adapter than you have in your console. For example, you cannot run a program that requires CGA graphics on an MDA card or run a program that requires full VGA capabilities on a CGA card. Note, however, that many DOS applications can sense the type of display adapter you have and adapt themselves to your hardware. For example, many DOS applications can display color and bit-mapped graphics when run on CGA or VGA cards and still run correctly (without displaying color) on an MDA card.

In general, DOS Services automatically senses the type of display adapter your console uses and can run any DOS applications that are compatible with that adapter.

The console also supports the UNIX MultiScreen mechanism (also sometimes called *virtual terminals*), which means you can run multiple display-oriented DOS or UNIX processes at once, each in its own screen. Refer to Chapters 2 and 6 and to your UNIX documentation for more information on the MultiScreen feature.

## Using EGA cards

---

EGA cards have several characteristics and optional configurations that do not apply to other display adapters. The following sections describe these characteristics and options.

### Prerequisites to using EGA cards

---

You can run applications that use EGA video modes on your console only if you have an EGA card that is 100% compatible with the standard IBM EGA. Your SCO Open Desktop/SCO Open Server system must also have a DOS EGA image. There are many EGA cards available that are not 100% compatible with the IBM EGA. These cards do not work properly when you use them with applications that require an EGA card on the SCO system.

If your display uses an EGA card, be sure that the card is set up in pure EGA mode. Some EGA cards can be configured to emulate other display adapters or to switch into different modes automatically. EGA cards that are set up to emulate other cards or autoswitch between different modes may not work properly when you use DOS Services.

Even if your computer has an incompatible EGA card, you can still run DOS applications. However, only monochrome and CGA modes are usable. When you use an incompatible EGA card, DOS Services treats your display as a CGA display by default.

Consult your system administrator if you are uncertain whether your the SCO system includes an IBM-compatible EGA card and has a DOS EGA image.

## Using EGA “fast” and “reliable” modes

When you run a DOS application on an IBM-compatible EGA card, DOS Services can run the application in *fast* EGA mode or in *reliable* EGA mode. DOS Services uses fast EGA mode by default.

Performance in fast EGA mode is faster than in reliable EGA mode. In fast EGA mode, however, your display can become corrupted when you use the UNIX MultiScreen feature to switch to another screen and then switch back to your DOS application. More specifically, your DOS screen may be corrupted under the following circumstances:

- The DOS application is in the middle of switching video modes at the moment you switch screens.
- The DOS application is actively drawing graphics on your display when you switch screens.
- The DOS application is writing directly to the EGA video registers when you switch screens.

If you do not use the UNIX MultiScreen feature while running DOS, you do not need to be concerned about these limitations. If you do use the UNIX MultiScreen feature, you can minimize the risk of corrupting your display by switching screens only when your application is not changing the DOS screen. You can also completely avoid these limitations by using the DOS Services reliable EGA option.

Reliable EGA mode avoids the limitations of fast EGA mode, but it causes DOS to run slower. To use reliable EGA mode, start DOS with the `+arega` option. For example, to start the DOS environment in reliable EGA mode, type the command:

```
$ dos +arega
```

The DOS environment starts and the DOS `C:` prompt appears as usual, and you can run any DOS commands or applications.

You can also start a specific application in reliable EGA mode from the UNIX shell. For example, to start an application called `COLORAMA`, you could issue the command:

```
$ dos +arega colorama
```

Chapter 5 describes how you can use the `dosopt` command to configure permanently the DOS environment or specific DOS applications to use reliable EGA mode.

## ***DOS processes that suspend when their screens are not visible***

---

When you run a DOS application that requires high-resolution graphics mode on an EGA card, DOS Services suspends the application if you use the UNIX MultiScreen feature to switch from the DOS screen to another screen. When you switch back to the DOS screen, the application resumes execution from the point where it was suspended.

This behavior is the same whether you use fast EGA mode or reliable EGA mode.

## ***Using VGA cards***

---

DOS Services is compatible with IBM VGA cards and can also run DOS applications correctly on many VGA cards made by other manufacturers. By default, DOS Services can use only the standard IBM video modes even if you have a VGA card that has nonstandard (extended) video modes. Your system administrator can create a new DOS image that enables you to use your VGA card's extended modes. See Chapters 2 and 6 or consult your system administrator for further information.

If your display uses an VGA card, be sure that the card is set up in pure VGA mode. Some VGA cards can be configured to emulate other display adapters or to switch into different modes automatically. VGA cards that are set up to emulate other cards or autoswitch between different modes may not work properly when you use DOS Services.

When you run a DOS application that requires high-resolution graphics mode on an VGA card, DOS Services suspends the application if you use the UNIX MultiScreen feature to switch from the DOS screen to another screen. When you switch back to the DOS screen, the application resumes execution from the point where it was suspended.

## ***Using PC scancode terminals***

---

PC scancode terminals emulate the keyboard layout and 25-line monochrome screen display of a standard PC and generate PC scancodes rather than ASCII codes. These terminals are connected to the DOS Services system via the UNIX serial driver. PC scancode terminals include Wyse Technology's WY-60 and Kimtron Corporation's Model KT-70/PC. You can run any monochrome DOS applications on a PC scancode terminal with the following limitations:

- You cannot run programs that display bit-mapped graphics.

- The UNIX MultiScreen (virtual terminal) feature is not available. When you start a DOS environment, you can return to the UNIX environment only by exiting from DOS (by typing **quit**, for example) or by forking a new UNIX shell.

## Using serial terminals

---

Serial terminals have the same limitations as PC scancode terminals. In addition, they typically have the following limitations:

- They can display only 24 of the 25 lines that DOS normally displays on a personal computer. The DOS Services **SHIFT SCREEN** terminal control code allows you to see either the top or bottom 24 lines.
- Many serial terminals cannot display the full IBM PC character set used by DOS. DOS Services maps characters that cannot be displayed to similar ASCII characters or to a period (.).

Some serial terminals can display the entire IBM PC character set. If you use the correct **TERM** definition for these terminals, DOS Services displays all PC characters as they would be displayed on a conventional DOS computer. (Refer to the *Multiuser Guide* for further information on setting up serial terminals.)

- The keyboards on most serial terminals are missing important keys such as **<Alt>**, **<PgUp>**, and **<PgDn>** that exist on PC keyboards. DOS Services lets you simulate these keys with combinations of standard ASCII keyboard keys. This release of DOS Services defines key sequences for the complete standard 101- and 102-key PC keyboards.

## Serial terminal keyboard mapping

---

PC keyboards contain keys not commonly available on standard serial terminals. Because DOS applications assume the presence of these keys, DOS Services supplies a method of simulating the missing PC keys using an ASCII keyboard. The following table shows the escape sequences that simulate these keystrokes on serial terminals.

| PC Key<br>to Simulate       | Escape Sequence |
|-----------------------------|-----------------|
| <b>The 12 Function Keys</b> |                 |
| ⟨F1⟩                        | ⟨Esc⟩ 1         |
| ⟨F2⟩                        | ⟨Esc⟩ 2         |
| ⟨F3⟩                        | ⟨Esc⟩ 3         |
| ⟨F4⟩                        | ⟨Esc⟩ 4         |
| ⟨F5⟩                        | ⟨Esc⟩ 5         |
| ⟨F6⟩                        | ⟨Esc⟩ 6         |
| ⟨F7⟩                        | ⟨Esc⟩ 7         |
| ⟨F8⟩                        | ⟨Esc⟩ 8         |
| ⟨F9⟩                        | ⟨Esc⟩ 9         |
| ⟨F10⟩                       | ⟨Esc⟩ 0         |
| ⟨F11⟩                       | ⟨Esc⟩ ;         |
| ⟨F12⟩                       | ⟨Esc⟩ :         |

|                           |         |
|---------------------------|---------|
| <b>The Numeric Keypad</b> |         |
| 4/←                       | ⟨Esc⟩ f |
| 6/→                       | ⟨Esc⟩ g |
| 8/↑                       | ⟨Esc⟩ t |
| 2/↓                       | ⟨Esc⟩ v |
| 7/⟨Home⟩                  | ⟨Esc⟩ h |
| 9/⟨PgUp⟩                  | ⟨Esc⟩ p |
| 3/⟨PgDn⟩                  | ⟨Esc⟩ q |
| 0/⟨Ins⟩                   | ⟨Esc⟩ i |
| ./⟨Del⟩                   | ⟨Esc⟩ d |
| 1/⟨Enter⟩                 | ⟨Esc⟩ e |
| 5                         | ⟨Esc⟩ . |
| ⟨Num Lock⟩                | ⟨Esc⟩ n |
| /                         | ⟨Esc⟩ & |
| *                         | ⟨Esc⟩ * |
| -                         | ⟨Esc⟩ - |
| +                         | ⟨Esc⟩ + |
| ⟨Enter⟩                   | ⟨Esc⟩ ^ |

|                                     |         |
|-------------------------------------|---------|
| <b>Shift, Control, and Alt Keys</b> |         |
| ⟨Left Shift⟩                        | ⟨Esc⟩ z |
| ⟨Right Shift⟩                       | ⟨Esc⟩ / |
| ⟨Left Ctrl⟩                         | ⟨Esc⟩ c |
| ⟨Right Ctrl⟩                        | ⟨Esc⟩ C |
| ⟨Left Alt⟩                          | ⟨Esc⟩ a |
| ⟨Right Alt⟩                         | ⟨Esc⟩ A |

| PC Key<br>to Simulate                   | Escape Sequence |
|-----------------------------------------|-----------------|
| <b>Other Special Keys</b>               |                 |
| ⟨PrtSc⟩/⟨SysReq⟩                        | ⟨Esc⟩ r         |
| ⟨Scroll Lock⟩                           | ⟨Esc⟩ s         |
| ⟨Pause⟩/⟨Break⟩                         | ⟨Esc⟩ l         |
| ⟨CapsLock⟩                              | ⟨Esc⟩ k         |
| ⟨SysReq⟩                                | ⟨Esc⟩ =         |
| ⟨Esc⟩                                   | ⟨Esc⟩ ⟨Esc⟩     |
| 102nd key on<br>international keyboards | ⟨Esc⟩ <         |
| <b>10 Cursor Keypad Keys</b>            |                 |
| ⟨Ins⟩                                   | ⟨Esc⟩ I         |
| ⟨Home⟩                                  | ⟨Esc⟩ H         |
| ⟨PgUp⟩                                  | ⟨Esc⟩ P         |
| ⟨Del⟩                                   | ⟨Esc⟩ D         |
| ⟨Enter⟩                                 | ⟨Esc⟩ E         |
| ⟨PgDn⟩                                  | ⟨Esc⟩ Q         |
| ↑                                       | ⟨Esc⟩ T         |
| ←                                       | ⟨Esc⟩ F         |
| ↓                                       | ⟨Esc⟩ V         |
| →                                       | ⟨Esc⟩ G         |
| <b>Special Function Escape Sequence</b> |                 |
| All shifts released                     | ⟨Esc⟩ (Space)   |

To simulate PC keys listed in the left column of the chart, press keys listed in the "Escape Sequence" column of the chart. For example, to simulate the ⟨Home⟩ key on an ASCII keyboard, press:

⟨Esc⟩ h

To simulate the ⟨F1⟩ function key on an ASCII keyboard, press:

⟨Esc⟩ 1

In a DOS environment, the PC keyboard's ⟨Ctrl⟩, ⟨Alt⟩, and ⟨Ctrl⟩ keys are usually used *in conjunction with* another keyboard character. To simulate these multiple-keystroke key combinations on a serial terminal, enter the escape sequence for the first key (⟨Esc⟩ c, ⟨Esc⟩ a, or ⟨Esc⟩ z), and then enter the escape sequence for the second key.

For example, to simulate the <Ctrl> <F1> key sequence, you would first press:

<Esc> z

to simulate the <Left Shift> key; then press:

<Esc> 1

to simulate the <F1> function key.

The <Ctrl>, <Alt>, and <Ctrl> key simulations all work in this manner. The escape sequences for these three keys send the “key down” (make) code to DOS. From the point of view of DOS, the key you are simulating remains down until you enter a keystroke that *does not* simulate <Ctrl>, <Alt>, or <Ctrl>.

For example, to simulate the common <Ctrl><Alt> <Del> reboot key combination from an ASCII keyboard, enter:

<Esc> c <Esc> a <Esc> d

In this example, <Esc> c simulates a depressed *and held* <Ctrl> key, <Esc> a simulates a depressed *and held* <Alt> key, and <Esc> d simulates the <Del> key. Since <Esc> d is the first key combination in this example that does *not* simulate <Ctrl>, <Alt>, or <Ctrl>, DOS Services sends to DOS both the simulation of the <Del> key *and* the “key up” (break) code for all three keys (<Ctrl>, <Alt>, and <Del>).

If the key combination you want to simulate ends with <Ctrl>, <Alt>, or <Ctrl>, you must end it with the <Esc> <Space> sequence, which sends the key up (break) code for any <Ctrl>, <Alt>, or <Ctrl> keystrokes currently in effect. For example, to simulate the <Ctrl><Alt> key combination used to access the Sidekick main menu, enter:

<Esc> c <Esc> a <Esc> <Space>

The escape sequence <Esc> <Esc> sends a real DOS escape character from an ASCII keyboard.

**NOTE** If your serial terminal has programmable function keys, you may set them up to generate single- or multiple-keystroke escape sequences that you use frequently. See the manual that accompanies the terminal for procedures.

In some cases, the function keys and arrow keys on a serial terminal are mapped to generate the proper PC-specific scancodes. This depends on the TERM setting for the type of terminal used.



## Console and terminal control codes

Key combinations called *control codes* allow you to perform several useful functions on your console, PC scancode terminal, or ASCII terminal. These functions are necessary and work only when you are in the DOS environment or running a DOS program. The following chart lists the key combinations you use to perform these functions:

| Function        | ASCII Terminal Control Keys | PC Scancode Terminal Control Keys | Console Control Keys |
|-----------------|-----------------------------|-----------------------------------|----------------------|
| KILL DOS        | ⟨Esc⟩ ⟨Ctrl⟩K               | ⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩K               | ⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩K  |
| DOS DEVICE INFO | ⟨Esc⟩ ⟨Ctrl⟩i               | ⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩-i              | ⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩i  |
| FORK SHELL      | ⟨Esc⟩ ⟨Ctrl⟩F               | ⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩-F              | N/A                  |
| SHIFT SCREEN    | ⟨Esc⟩ ⟨Ctrl⟩U               | ⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩U               | N/A                  |
| REDRAW SCREEN   | ⟨Esc⟩ ⟨Ctrl⟩R               | ⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩R               | N/A                  |

Use these control codes as follows:

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KILL DOS        | If a DOS process becomes irrecoverably hung (for example, when ⟨Ctrl⟩⟨Break⟩ and ⟨Ctrl⟩⟨Alt⟩ ⟨Del⟩ fail), use this key combination to kill DOS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| DOS DEVICE INFO | When you run many different DOS environments, it is easy to forget which devices are attached. The <i>DOS Device Information Popup Window</i> displays all the devices available to your current DOS session.                                                                                                                                                                                                                                                                                                                                                                                                                |
| FORK SHELL      | FORK SHELL starts a new UNIX shell from the DOS environment or a DOS application, but it does not use the standard UNIX MultiScreen (virtual terminal) mechanism. You therefore cannot use the standard UNIX MultiScreen key sequences to switch between your DOS session and UNIX shell. You can, however, start any UNIX or DOS process—including a new DOS environment—in the UNIX shell. Terminate the UNIX shell and return to your original DOS session just as you would a UNIX shell started from a UNIX prompt: press ⟨Ctrl⟩D or type <i>exit</i> .<br><br>The FORK SHELL function is not available on the console. |
| SHIFT SCREEN    | On a 24-line display, this control code causes the screen to shift from displaying the lower 24 lines (the default) to displaying the upper 24 lines. Pressing this control                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

code again shifts the screen back to the default display. This control code has no effect on a 25-line display or PC scancode display.

**REDRAW SCREEN**      If the DOS display is overwritten or corrupted, use this control code to redraw the screen. This control code has no effect on the console.

Note that on some serial terminals, the key sequence <Ctrl><Esc> accomplishes the same thing as just pressing <Esc>. Therefore, you can use the same <Ctrl><Esc> sequence on these ASCII terminals to introduce terminal control codes as you do on PC scancode terminals and the console.

## Appendix B

# Terminfo guide

---

This appendix is divided into the following sections:

- “Basic concepts” describes the **terminfo** file, the terminals it serves, and the special key sequences used to activate its services.
- “Output on standard terminals” and “Input on standard terminals” describe the **terminfo** items used for standard terminals.
- “About PC scancode terminals” discusses the process of switching between standard mode and PC scancode mode on a PC terminal.
- “Output on PC scancode terminals” and “Input on PC scancode terminals” describe some **terminfo** items that allow DOS Services to take advantage of the extra capabilities of PC scancode mode.
- “PC-character-to-ASCII conversions” contains the PC-character-to-7-bit-ASCII conversions that are built into DOS Services.
- “**terminfo** escape sequences” displays the **terminfo** key definitions and the standard escape sequences that DOS Services uses to simulate a 101/102-key PC keyboard.

The primary requirement for DOS to run on a terminal is a good **terminfo** description. Therefore, most of this appendix describes the standard **terminfo** items used by DOS Services. In addition, there are special **terminfo** items used by DOS Services to enhance the use of DOS on a terminal.

This appendix is not a tutorial on how to construct **terminfo** descriptions for standard terminals; for that, refer to the appropriate UNIX documentation. We recommend that you study the **terminfo**(M) man page for information about the structure and use of **terminfo** source files.

## Basic concepts

---

### *terminfo*

---

**terminfo** is the UNIX terminal description utility. It includes a set of descriptions of terminal capabilities and a set of routines that use those descriptions. **terminfo** files are compiled databases that describe the capabilities of the terminals attached to the system and how those terminals perform their various operations.

This appendix describes how DOS Services uses **terminfo** entries to run DOS sessions on serial terminals. It also presents various ways of enhancing your existing **terminfo** files for better DOS support.

DOS programs expect their keyboard input in the form of PC scancodes and write characters to the screen in a PC character set. Serial terminals use entirely different codes for their keyboard input and their screen output. Therefore, running DOS on serial terminals requires several conversions. The terminal's keyboard input must be converted to the PC scancodes that DOS programs expect. The PC character set output by the DOS program must be converted to whatever code sequences are used to position and display characters on that particular terminal's screen.

DOS Services uses two processes that do this conversion. The **tkbd** process converts the input and the **crt** process converts the output. These processes use the **terminfo** descriptions for the terminal to do their processing. Besides hardware and the standard components already provided by the DOS Services system, the only additional item you need to run DOS on a terminal is a good, complete **terminfo** description.

### *terminfo files*

---

Each **terminfo** file is a database of terminal capability descriptions. Conceptually, the descriptions can be split into two categories: *terminal input* and *terminal output*.

**terminfo** descriptions use two methods of describing each terminal capability. *Variable* is the name the programmer uses to access the **terminfo** capability. *Capname* (capability name) is a shortened name used in the text of the database. It is used primarily by people updating the database.

For terminal input, the database describes how keyboard input is to be interpreted. These descriptions are called *key definitions* in this appendix and have **terminfo** variable names beginning with **key\_**. For example, the **terminfo** **key\_f1** string (capname **kf1**) holds the characters that are generated when function key <F1> has been pressed.

For terminal output, the database describes how to display information on the screen. For example, it describes how to display a character in reverse video.

In addition, `terminfo` also describes some special commands to control the operation of the terminal. One example is a command to turn on a special scrolling mode.

## Standard and PC scancode terminals

---

There are two basic varieties of serial terminals, *standard* terminals and *PC scancode* terminals.

Standard terminals are the type normally used with UNIX systems. These standard terminals display several different character sets. Some display only 7-bit ASCII characters. Others display 8-bit character sets, such as the ISO Latin-1 character set. Some terminals can display multiple character sets (though usually only one at a time).

Note that the 7-bit ASCII character set is a subset of most character sets used by standard terminals, including PC character sets. The principal characteristic of a "standard" terminal is that the keyboard sends the same character codes that the terminal displays (or a subset). For example, on an ASCII terminal, when you press the "A" key, the ASCII code for "A" is transmitted. Shift keys, control keys, or *meta* keys do not send codes. They merely modify the code that is sent when a regular key is pressed. The code is sent only when the key is pressed. No code is sent when the key is released.

Every PC scancode terminal is based on a standard terminal. Thus they have many of the same qualities, with the PC feature as a special mode of the terminal. When in PC scancode mode, the terminal sends PC scancodes instead of the normal keyboard codes. A scancode is sent when a key is pressed, and another is sent when the key is released. All PC scancode terminals can display the entire PC character set, at least when in PC scancode mode.

The PC scancode mode is turned on and off with a command sent from the server system to the terminal. It is usually the responsibility of the program controlling the terminal (such as DOS Services) to put the terminal into scancode mode when starting and put it back into standard mode when finished.

Note that a terminal can have the ability to display the PC character set but still not be considered a "PC terminal." A "PC terminal" must be able to send PC scancodes as well as display a PC character set. This is why they are called *PC scancode* terminals.

## Escape sequences

---

An *escape sequence* is a pattern of bytes used to send some sort of special command. Originally the term referred to sending the ASCII escape character followed by a byte sequence used as a command addressed to hardware. This usage has broadened over time. Current usage includes special command sequences sent either by programs, or from the keyboard, by users.

The DOS Services implementation of the escape sequence refers to a pattern of key presses trapped by DOS Services before it reaches the user's application program. The application may or may not be terminated as a result. DOS Services may act upon the key sequence itself or pass it on to the DOS program as keyboard input.

The escape sequence is so called because the first key is often <Esc>. This initial keystroke is configurable, but even if it has been changed, these special patterns of keystrokes are still referred to as "escape sequences." DOS Services uses escape sequences for two purposes: key emulation and special actions.

DOS programs expect input from certain special DOS keys that may not be physically present on a particular terminal's keyboard. Function keys and arrow keys are examples. If the key does not physically exist, you can emulate that key's action for DOS by pressing a sequence of keys that do exist. DOS Services intercepts this pattern and translates it into the codes expected by the DOS program. This is referred to as a *key emulation escape sequence*. It is usually <Esc> plus another key.

While running a DOS program, you sometimes need to communicate directly with DOS Services itself, to request some special action. You may want the screen redrawn or to call up the DOS Device Information Popup Window. DOS Services provides special escape sequences for this, and this appendix calls them *special action escape sequences*.

## The DOS Services lead-in sequence

---

The *lead-in sequence* is one or more keystrokes that initiate an escape sequence. DOS Services uses both one- and two-character lead-ins. In some computing environments, the lead-in is the <Esc> key alone. In others it is <Ctrl><Esc>. The lead-in sequence is configurable, and you can specify any keys you find convenient.

## Escape sequences used by DOS Services

---

ASCII and ISO terminal keyboards lack some PC keys, so key emulation sequences are needed for those terminals. The default setup emulates missing DOS keys with <Esc> plus a character. The default sequence for special actions is <Esc> plus a control character.

PC scancode terminals have all the PC keys, so key emulation sequences are not needed. They do, however, need to send requests for special actions to DOS Services and they do this with a two-key lead-in plus another key. The default is (Ctrl)(Esc) plus a control character.

DOS Services intercepts all keyboard input from the terminal. Most of it is passed on to the current DOS program as input. The DOS Services lead-in character triggers two different responses, based upon what comes next.

The lead-in character plus a standard ASCII character generally signals a key emulation. The codes for the emulated key are passed on to the DOS program as input. The lead-in sequence plus a control character ((Ctrl) held down while another key is pressed) generally signals a special action. This sequence is acted upon by DOS Services itself.

## Output on standard terminals

This section discusses **terminfo** items that DOS Services uses to display DOS characters on serial terminals.

The terminal must have the following minimum capabilities to support DOS:

| Variable name  | Capname |
|----------------|---------|
| cursor_address | cup     |
| clear_screen   | clear   |
| clr_eol        | el      |

This value must also be set:

| Variable name | Capname |
|---------------|---------|
| lines         | lines   |

Screen scrolling must be available. The first two capabilities listed below represent primitive, line-oriented capabilities that can be used if they are all that exist. The next three capabilities provide much better facilities. One group or the other must exist for DOS to use the terminal.

| Variable name | Capname |
|---------------|---------|
| insert_line   | il1     |
| delete_line   | dl1     |

| Variable name        | Capname |
|----------------------|---------|
| change_scroll_region | csr     |
| scroll_forward       | ind     |
| scroll_reverse       | ri      |

The following capabilities are used if they exist but are not required:

| Variable name         | Capname |
|-----------------------|---------|
| cursor_invisible      | civis   |
| cursor_normal         | cnorm   |
| cursor_up             | cuul    |
| cursor_down           | cud1    |
| cursor_left           | cub1    |
| cursor_right          | cuf1    |
| enter_ca_mode         | smcup   |
| exit_ca_mode          | rncup   |
| enter_am_mode         | smam    |
| exit_am_mode          | rmam    |
| enter_pc_charset_mode | smppch  |
| exit_pc_charset_mode  | rmpch   |

All but two of the **terminfo** items listed above are explained in your UNIX **terminfo** documentation. The **smppch** (enter\_pc\_charset\_mode) and **rmpch** (exit\_pc\_charset\_mode) capabilities are special command strings. They set the modes of the terminal when starting and finishing a DOS task. For example, if the terminal has special commands to enable and disable a PC character set, those commands would be put in **smppch** and **rmpch**. Other commands can also be put in these strings to be executed when starting and finishing a DOS task. Examples of this are commands that change the cursor shape to an underscore when starting DOS and change it back to normal when exiting DOS. Refer to “About PC scancode terminals” for more information about PC scancode mode terminals.

**NOTE** Commands that actually switch a terminal into and out of PC scancode mode *must not* be put into **smppch** and **rmpch**. There are separate commands for that. See **smsc** and **rmsc** in “About PC scancode terminals.”

The following **terminfo** commands can be executed (if they exist) when starting and ending a DOS session. They are listed in the order that they are used.



| Variable Name         | Capname            |
|-----------------------|--------------------|
| Starting:             |                    |
| enter_pc_charset_mode | smpch <sup>1</sup> |
| exit_am_mode          | rmam               |
| enter_ca_mode         | smcup              |
| clear_screen          | clear              |
| set_attributes        | sgr                |
| Ending:               |                    |
| change_scroll_region  | csr                |
| cursor_address        | cup                |
| clr_eol               | el                 |
| exit_ca_mode          | rmcup              |
| enter_am_mode         | smam               |
| exit_attribute_mode   | sgr0               |
| cursor_normal         | vnorm              |
| exit_pc_charset_mode  | rmpch <sup>1</sup> |

Other important settings are those that describe the sequences for controlling the various display modes, such as Bold, Reverse, Underline, and Blink. The following is a list of the **terminfo** display mode commands used when displaying a DOS session. These are not required, but, if they are available, they make the display look much more like a real DOS console screen. Of these, **sgr** (**set\_attributes**) is the most important. In fact, a well-written **sgr** entry can make all the difference between a good DOS display and a substandard one.

| Variable Name        | Capname |
|----------------------|---------|
| set_attributes       | sgr     |
| enter_reverse_mode   | rev     |
| enter_secure_mode    | invis   |
| enter_underline_mode | smul    |
| enter_bold_mode      | bold    |
| enter_blink_mode     | blink   |
| exit_attribute_mode  | sgr0    |

Note that real PC monochrome text has more display modes than some terminals. Some terminals have only one brightness level. Others have brightness levels that do not match those of a PC monochrome display. (PC monochrome has "normal" and "high intensity" levels; some terminals have only "dim" and "full" brightness levels.) For this reason it is sometimes difficult to choose which terminal mode best conveys the information the DOS program is trying to communicate.

1. Remember that, despite their names, these commands are *not* used to switch the terminal into and out of the PC scancode mode. Refer to the descriptions of **smsc** and **rmsc** under "Switching Commands" in "About PC Scancode Terminals."

PC monochrome characters have four mutually exclusive display modes:

- Normal (light character on dark background)
- Reverse video (dark character on light background)
- Invisible
- Underline

In addition, the characters can be high-intensity or blinking. Thus there are 16 possible ways to display a character (including four invisible ones). The only way all these combinations can be correctly displayed is with the **sgr** capability. The **sgr** string is a parameterized string. **sgr** and parameterized strings are discussed in detail in the **terminfo(4)** manual page. Please refer to that document for more information.

## *Special terminfo descriptors*

---

We have discussed standard **terminfo** descriptors, plus two special ones, **smprch** and **rmprch**. There are some additional special **terminfo** descriptors used by DOS Services to enhance the operation of DOS on terminals. The next two sections discuss two of the most important, **dispc** and **pctrm**. Additional special descriptors are discussed later, in the sections covering terminal input and the PC terminal.

| Variable Name               | Capname      |
|-----------------------------|--------------|
| <b>pc_term_options</b>      | <b>pctrm</b> |
| <b>display_pc_character</b> | <b>dispc</b> |

## *pctrm*

---

The **pctrm** string is used to set special input and output processing options for DOS Services. The output related options are:

- **LRC\_OK**
- **ISO\_TERM**
- **CS\_SW\_INDEPENDENT**
- **PCMODECP**

**LRC\_OK** is set when it is OK to write to the lower right corner of the display. Many terminals incorrectly scroll when that location is written to, even when the automatic scroll mode is turned off. Consequently, by default, DOS Services does not write to this location. If the terminal does not have this problem, set **LRC\_OK** in the **pctrm** string.

The NLS (National Language Support) features of DOS Services provide support for special character sets. (The prime example is ISO 8859.) DOS Services offers special facilities for terminals that can display these character sets. For this type of terminal, put `ISO_TERM` into the `pctrm` string.

When you find it inconvenient to modify the `terminfo` to set this option, you can set a UNIX environment variable `ISO_TERM` instead. The section “How PC characters are displayed on ISO terminals,” later in this appendix, provides details on how this is done.

When an ISO terminal has a special mode that enables it to display a PC character set, the options `CS_SW_INDEPENDENT` and `PCMODECP` can be used. The section “Combined ISO and PC character set terminals,” later in this appendix, discusses these two special options.

Any combination of options is valid within the `pctrm` string; the options can be placed in any order. Every option in the `pctrm` string must be delimited by leading and trailing colons (:). The following examples show the valid combinations of `LRC_OK` and `ISO_TERM`.

```
pctrm=:LRC_OK:ISO_TERM:,
pctrm=:ISO_TERM:LRC_OK:,
pctrm=:ISO_TERM:,
pctrm=:LRC_OK:,
```

## *dispc*

---

By default, DOS Services uses an internal table to map each PC character to a “matching” ASCII character that the terminal can display. Many extended PC characters that have no good ASCII match are displayed simply as “\*”. See “PC-character-to-ASCII-conversions,” at the end of this appendix, for these built-in conversions.

The `terminfo` string `dispc` can be used to override DOS Services’s built-in table. `dispc` can be used in one of two ways. It can either contain the name of an alternate conversion table (for example, `dispc=/usr/lib/dispc/myterminal`), or it can be a parameterized string that describes the *rules* for displaying PC characters.

If you have a serial terminal capable of displaying *some* of the PC characters (but not all), or if you want to redefine some of the built-in character conversions, the first form of `dispc` suits your needs best. On the other hand, for a PC scancode terminal or any terminal that can display most of the PC characters, the second form of `dispc` is more efficient.

The next section describes the first form of **dispc**. The section "Output on PC scancode terminals" discusses the parameterized form of **dispc**.

## *How PC characters are displayed on terminals*

---

The first form of **dispc** string is used to specify a file containing a PC-character-to-terminal-code conversion table. This is known as a PCT file.

The PCT file is a normal UNIX text file, with each line defining a single PC-character-to-terminal-character conversion. In a PCT file each line is composed of fields, separated by space or tab characters. Any field that starts with `"/*"` starts a comment. In addition, lines that have a `"#"` as the first character in the first field are comments. All comments end at the end of the line. Each line can have a maximum of eight usable fields.

The first field is the PC character code to be displayed. The following fields (up to seven) are the codes to be sent to the terminal to display this character. The order of the lines is not important. If a particular PC character conversion is not specified in the file, the built-in conversion is used for that PC character. Each field can be specified in four ways: it can be a 7-bit ASCII character, a decimal number, a hex number, or an octal number.

A decimal number is represented as a number 0 to 255 with a period at the end (for example `"129."`). A hex number is represented as a number 0 to FF with an `"h"` at the end (for example `"A8h"`). An octal number is represented as a number 0 to 377 with an `"o"` at the end (for example `"267o"`). Here are a few sample PCT lines:

```
z z /* This converts 'z' to 'z'
 /* (no conversion needed).
e1h 33o 1. B /* e1 is the beta symbol, and the beta is
 /* displayed with an escape (33 octal)
 /* followed by binary '1' which in turn is
 /* followed by the ASCII character 'B'.
b3h 27. b3h /* To display the b3 (vertical line)
 /* character, it must be preceded with an
 /* escape (27 decimal).
```

## *How PC characters are displayed on ISO terminals*

---

DOS Services supports NLS automatic character set conversion. Each character in the PC character set being used by the DOS session is converted to the closest character in the ISO character set. The ISO character is then displayed on the terminal.

DOS Services automatically keeps track of the current PC code page of the DOS session. This allows it to convert from the current PC code page to the terminal's character set, even when the PC code page is changed in the middle of a session.

Not all NLS character sets are supported. Factory-supplied NLS character set support includes (at a minimum) ISO 8859 (the Latin-1 alphabet), plus PC code pages 437 and 850. Other character sets may also be supported. Refer to your Release Notes for a full list of supported character sets.

*Code set* is the UNIX term for the character set your terminal uses. DOS Services determines which code set your terminal uses via the **LANG** environment variable or by reading **/etc/default/lang** if it exists. For details about the **LANG** environment variable and instructions on how it is used to specify the code set, refer to Chapter 7 of this book and to your UNIX documentation.

DOS Services automatically keeps track of the current PC code page. Provided that both your terminal's code set and the PC code page in use are supported, translation between the two is automatic.

For example, suppose the PC character to be displayed is code number 0x82 in code page 850 (é). DOS Services translates this into the code number for "é" in the ISO 8859 character set your terminal can display.

This automatic character set conversion is enabled whenever **ISO\_TERM** has been included in the **terminfo pctrm** string or the **ISO\_TERM** environment variable is set. **ISO\_TERM** does not need to be set to anything in particular. It merely needs to exist. From the Bourne or Korn shell, use the following command to set it:

```
ISO_TERM=
export ISO_TERM
```

When an exact conversion for a character is not possible, DOS Services uses the closest match it can find in its own built-in conversion table. If a specific conversion has been specified with **dispc**, that transformation is used instead.

## *Combined ISO and PC character set terminals*

---

As a very special case, DOS Services also supports those terminals that display both an ISO code set and a PC code page. Whenever the DOS process is using a code page that matches the code page supported by the terminal, the terminal is switched to PC character set mode. DOS Services sends the same output to the terminal it would send to any terminal that can display that PC character set.

When the DOS process switches to a PC code page other than the one the terminal supports, the terminal is switched to ISO mode. DOS Services then sends the same screen output it would send to a normal ISO terminal.

There are several restrictions. The **smpch** and **rmpch** terminfo items must be specified. They must contain only the commands that switch the terminal from ISO code set display to the PC code page display. The terminal cannot change its mode in any respect other than the character set being displayed. It cannot use a PC scancode mode that changes how the keyboard operates, or changes how the character attributes are set. **dispc** must also be specified, so that DOS Services knows how to display PC characters when in PC character mode.

By default, it is assumed that code page 437 is used by the terminal when in PC code page mode. If another code page is to be used, the string, **PCMODECP=pcnnn** is set in **pctrm**. This specifies that code page *nnn* is used by this terminal.

To enable ISO-to-PC-character-set switching, the **pctrm** string must contain both **ISO\_TERM** and **CS\_SW\_INDEPENDENT**. The **PCMODECP=pcnnn** string is optional. For example, if code page 850 is to be used, **pctrm** would be set like this:

```
pctrm=:ISO_TERM:CS_SW_INDEPENDENT:PCMODECP=pc850;
```

When the DOS session is using a code page that matches the PC code page the terminal can display, the terminal is switched into that mode with **smpch**. **dispc** is used to display the characters. When switching from that code page to another, **rmpch** is used to switch the terminal back to ISO display mode.

## *Input on standard terminals*

---

On standard terminals, the keyboard codes coming from the terminal are either 7-bit ASCII or 8-bit ISO 8859 (the Latin-1 character set). DOS programs assume that their input is coming from a PC keyboard and expect it in the form of PC scancodes. To emulate a DOS keyboard, DOS Services translates serial terminal keyboard input codes into the scancode *make* and *break* signals sent by the corresponding keys on a PC keyboard.

DOS Services uses internal tables to translate the ASCII or ISO input characters into the equivalent PC character. Then the PC characters are sent to DOS as PC keyboard make and break scancodes.

To translate the keyboard input, DOS Services needs to know what character set is being used on the terminal. It gets this information from the UNIX environment variable **LANG**. (Refer to Chapter 7 for more information about the **LANG** environment variable and code sets.) DOS Services keeps track of the current DOS code page character set and adjusts its translation from the UNIX code set as the DOS code page changes.

When the character code coming from the terminal corresponds to an unshifted PC keyboard key, the make and break for that key are simply delivered to the DOS process. For the "A" key, this would be scancodes 1E for the make and 9E for the break.

When the keyboard code coming from the terminal corresponds to a *shifted* PC key (either regular <Shift> or <Ctrl><Shift>), more is required. The make and break signals for that key are preceded by a make of the appropriate <Shift>-type key and followed by a break of that <Shift> key. For <Ctrl>A, this would be four separate codes: 1D for the make of the <Ctrl> key, 1E for the make of the "A" key, 9E for the break of the "A" key, and 9D for the break of the <Ctrl> key. This behavior is called *auto-break*. Nothing special needs to be done to cause the key break to happen.

Where the DOS program requires input from PC keys that are not physically present on the terminal keyboard, DOS Services provides key emulation escape sequences. For example, the sequence to simulate the <SysReq> key, is <Esc>=. You send keyboard codes equivalent to ASCII Esc and the ASCII equal sign. DOS Services translates this into the make and break scancodes for the <SysReq> key and sends those codes to the DOS process.

The tables in Appendix A list the default key emulation escape sequences. All these keys are auto-break; when the sequence is received by DOS Services, the make and break codes for the appropriate key are automatically sent to DOS.

The <Shift>-type keys (<Shift>, <Ctrl>, and <Alt> keys) are *semi-auto-break*. When received, the make code for the <Shift>-type key is sent. The break code is delayed until one of two things occurs:

- An auto-break occurs for another key.
- The release-all-shifts sequence is received.

The release-all-shifts sequence allows you to force the release of the <Shift>-type keys <Shift>, <Ctrl>, or <Alt>. The default sequence is <Esc><Space>. This causes breaks for all made <Shift>-type keys, so that isolated <Shift>-type key actions may be simulated.

For example if you need to simulate the pressing and releasing of the left and right <Shift> keys, you would use this key-stroke sequence: <Esc>, Z, <Esc>, I, <Esc>, <Space>.

## Default "key emulation" escape sequences

Each key emulation escape sequence is made up of two keyboard codes. By default the lead-in is the ASCII ESC code (hex 1b). This is followed by an ASCII character code. See "Serial terminal keyboard mapping" (page 345) for a list of the default key emulation escape sequences that DOS Services uses to simulate DOS keys not physically present on the terminal keyboard.

## Default “special action” escape sequences

---

In addition to the key emulation escape sequences, there are *special action escape sequences* that are intercepted by and acted on by DOS Services itself. The following table lists these.

| Press (Esc) plus | To Activate                |
|------------------|----------------------------|
| <sup>^</sup> U   | See top or bottom 24 lines |
| <sup>^</sup> F   | Fork a new shell           |
| <sup>^</sup> R   | Redraw screen              |
| <sup>^</sup> K   | Kill DOS                   |
| <sup>^</sup> Z   | Suspend DOS <sup>2</sup>   |
| <sup>^</sup> I   | Display DOS Devices        |

Again, each special action escape sequence is a grouping of two codes: the lead-in sequence followed by a control-key combination. Remember that (Ctrl) is a (Shift)-type key. (Ctrl) plus a second key press is a single event. The (Ctrl)-*second-key* combination is sent as a single code.

The default lead-in on standard terminals is the ASCII ESC code (hex 1b). On PC scancode terminals, the default lead-in is (Ctrl)(Esc). The lead-in is followed by pressing (Ctrl) plus one of the letter keys listed in this table.

In the above table, <sup>^</sup>U stands for (Ctrl)U. Similarly, <sup>^</sup>F, <sup>^</sup>R, <sup>^</sup>K, <sup>^</sup>Z and <sup>^</sup>I stand for (Ctrl)F, (Ctrl)R, (Ctrl)K, (Ctrl)Z and (Ctrl)I, respectively.

## Using terminfo descriptors for key emulation

---

We have already described the escape sequences built into DOS Services for key emulation. The built-in key emulation escape sequences cover every PC key a DOS program expects and assume that the terminal has no such keys whatever.

Most terminals, however, have some of the same keys found on PC keyboards (such as the function and arrow keys). The code sequences they send when those keys are pressed are not standardized, but the keys are often physically present. Consequently, a well-constructed **terminfo** file for such a terminal already contains descriptors for whatever function and arrow keys the terminal has.

---

2. The “Suspend DOS” function is only available under a job control shell.



Whenever they exist, DOS Services uses these pre-existing descriptors, in addition to its own built-in key emulation escape sequences. If you are writing or editing a **terminfo**, you can add or modify its entries in any way you find convenient, and DOS Services uses those values in addition to its built-ins.

Thus, when working at a terminal equipped with a physical **<F11>** key, you can send **<F11>** to DOS Services in two ways. You can press the physical **<F11>** key on the keyboard or activate the built-in sequence with **<Esc>**;

DOS Services monitors the code sequences sent by the terminal constantly. Whenever a code sequence matches either its own built-in key emulation escape sequence or one of the key descriptions in **terminfo**, DOS Services sends the make and break codes for that key to the DOS process.

Refer to “**terminfo** escape sequences,” at the end of this appendix, for a comprehensive list of **terminfo** entries and how they can be mapped to PC keyboard keys. Wherever a PC key has no corresponding key on the terminal keyboard, and thus no corresponding entry in the **terminfo** file, an unused entry has been chosen to represent it. For example, the PC **<SysReq>** key has been mapped to the **key\_options (kopt)** entry. Where no pre-existing entry bears any relationship to the PC key, an entry for an unused function key has been used. If you create or customize your own **terminfo** files, use “**terminfo** escape sequences” as a guide.

## ***pctrm***

---

If you want to configure the terminal so that it uses *only* the built-in key emulation escape sequences and ignores any existing **terminfo** key descriptions, use the **pctrm** string. We have discussed its output-related options. The one input-related option is **NO\_TL**.

Whenever this keyword is in the **pctrm** string, DOS Services ignores the key descriptors in **terminfo**. This is useful when there are key descriptors that conflict with the built-in DOS Services escape sequences.

Sometimes it is easier to force DOS Services to use only the built-in sequences than to make the user resort to some new escape sequence. For example, if the **<F1>** key on the terminal transmits **<Esc>A**, this conflicts with the built-in escape sequence for the right **<Alt>** key. A new escape sequence for the right **<Alt>** key could be put into **terminfo**, but users might be more comfortable using all the default escape sequences.

In this example the user would not press the **<F1>** key (or any of the special keys). The user would press **<Esc>1** to simulate the **<F1>** key for DOS. All the other default escape sequences would also be used to simulate the special keys.

The following example shows how to set the NO\_TI option.

```
pctrm=:NO_TI,
```

## Changing the lead-in character

---

By default, the (Esc) key itself is used as the lead-in (first) character for the default escape sequences.<sup>3</sup>

Sometimes there are conflicts between the built-in escape sequences and other key combinations used by the terminal. When the terminal needs the (Esc) key for some other purpose, you can reconfigure the lead-in character. You specify another lead-in character with the **terminfo** `scesc` descriptor.

---

| Variable Name                | Capname            |
|------------------------------|--------------------|
| <code>scancode_escape</code> | <code>scesc</code> |

---

For example, `scesc=Z` sets the lead-in sequence to `Z`. `scesc=^X` sets it to (Ctrl)X.

## About PC scancode terminals

---

This section discusses the characteristics of PC terminals and how their special capabilities can be activated by the information in **terminfo**.

PC terminals are based on standard serial terminals, so most of the foregoing issues apply to PC terminals just as they do to standard terminals. PC terminals have a special mode, however. In PC mode they emulate a PC's Monochrome Display Adapter (MDA) and the PC scancode keyboard.

The keyboard has the same layout as a PC keyboard and can send PC keyboard scancodes. The screen can display all the same characters as an IBM-compatible MDA.

These terminals are usually in standard mode, acting as normal serial terminals for the UNIX system. When a DOS session is started, DOS Services switches the terminal into the PC scancode mode. When the DOS session is finished, DOS Services switches the terminal back to the standard mode. The following **terminfo** items are used when switching between standard and PC scancode modes.

---

3. Remember, an escape sequence must also be used for the (Esc) key itself. This sequence is (Esc)(Esc). To send the (Esc) key to DOS, you must type it twice.

| Variable Name       | Capname |
|---------------------|---------|
| enter_scancode_mode | smisc   |
| exit_scancode_mode  | rmisc   |
| user0               | u0      |
| user1               | u1      |
| user2               | u2      |
| user3               | u3      |
| pc_term_options     | pctrm   |

## Switching commands

The first two are the *switching commands*. Note the name similarity between these two commands and two commands described earlier. **smisc** and **rmisc** put the terminal into PC scancode mode and return it to standard mode. **smprch** and **rmprch**, described earlier, are command strings that specify the modes of the terminal when the DOS process starts and when it exits.

## Alternate flow control

Items **u0**, **u1**, and **u2** control the so-called XPC feature. Many PC scancode terminals offer these alternate flow-control characters for PC scancode mode.

Standard terminals use two characters for flow control, normally referred to as XON and XOFF. On standard terminals, they are usually keyed with (Ctrl)Q and (Ctrl)S. These characters cannot be used in scancode mode, so alternate flow-control characters are needed.

The **u0** string defines these scancode mode XPC flow-control characters. The first character in the string is the XON character and the second is XOFF. The **u1** string is the command sent to the terminal to enable the XPC flow control. **u2** disables it. If the terminal automatically uses XPC flow control when put into scancode mode, **u1** and **u2** do not need to be specified.

For example, the typical XPC characters are 0x65 for XON and 0x67 for XOFF. Since the number 0x65 is ASCII "e" and 0x67 is ASCII "g", the **u0** specification that defines these XPC characters is:

**u0=eg,**

## Serial line parameters

The **u3** string contains the command sent to the terminal to set the communication parameters: baud rate, number of bits, stop bits, and parity.

Whenever the terminal is in scancode mode, the serial line must be in 8-bit mode. Normally, the user sets up the serial line in such a way that it is always in 8-bit mode when using a PC terminal. In this case, the command is not needed.

The sole reason for this command is for those situations where the line has not been so configured. If the user forgets to put the line into 8-bit mode, the software can perform the service automatically. The line is placed in 8-bit mode when putting the terminal into scancode mode. The line is returned to 7-bit mode when the terminal exits scancode mode.

This is a **terminfo** parameterized string. For more information about parameterized strings, see the **terminfo(4)** manual page. The following table defines the parameters.

| Parameter | Use                            |
|-----------|--------------------------------|
| 1st       | Not used                       |
| 2nd       | Number of data bits (7 or 8)   |
| 3rd       | Number of stop bits (1 or 2)   |
| 4rd       | Parity (0=none, 1=odd, 2=even) |
| 5th       | Baud rate (1 thru 15 decimal)  |

Baud rate IDs 1 through 15 correspond to baud rates of 50 thru 38400. The baud IDs are drawn from the **CBAUD** field of **termio**. (They are defined in the file `/usr/include/sys/termio.h`). The following table defines the ID for each baud rate.

| Baud | ID | Baud | ID | Baud  | ID |
|------|----|------|----|-------|----|
| 50   | 1  | 300  | 7  | 9600  | 13 |
| 75   | 2  | 600  | 8  | 19200 | 14 |
| 110  | 3  | 1200 | 9  | 38400 | 15 |
| 134  | 4  | 1800 | 10 |       |    |
| 150  | 5  | 2400 | 11 |       |    |
| 200  | 6  | 4800 | 12 |       |    |

It may seem surprising that baud rate is included in a single command with the other communication parameters. This is because some terminals have only a single command that sets all of these parameters at once. Any time you change any one parameter, all parameters, including the baud rate, must be specified.

## ***pctrm***

The **pctrm** string defines special options for either a standard terminal or a PC scancode terminal. They are activated by inserting specific terms in the **pctrm** string.

One special option relates to switching PC terminals between standard and PC scancode modes. This option is:

**USE\_TERM**

This option is required when a terminal's PC scancode mode uses completely different commands than those used in standard mode.

One example is the WY-60 terminal. The Wyse **terminfo** descriptions for standard mode do not work when the terminal is in PC scancode mode. **USE\_TERM** is employed to specify an alternate **terminfo** name for use when the terminal is in PC scancode mode. The syntax of this option is **USE\_TERM=xxx** where **xxx** is the **terminfo** name to be used when in PC scancode mode. As with the other **pctrm** options, any combination and order of options is valid within the **pctrm** string. Every option in the **pctrm** string must be delimited by leading and trailing colons (:).

For example, suppose the **terminfo** name to be used when the Wyse is in PC scancode mode is **wy60-pc**. The **pctrm** string would be defined in the standard **terminfo** as:

**pctrm=:USE\_TERM=wy60-pc;**

## Sequence of use

The following table presents a list of the **terminfo** commands used (if they exist) when starting and ending a DOS session on a PC scancode terminal.

|           | Variable Name                | Capname      |
|-----------|------------------------------|--------------|
| Starting: |                              |              |
|           | <b>user3</b>                 | <b>u3</b>    |
|           | <b>enter_scancode_mode</b>   | <b>smsc</b>  |
|           | <b>user1</b>                 | <b>u1</b>    |
|           | <b>enter_pc_charset_mode</b> | <b>smpch</b> |
|           | <b>exit_am_mode</b>          | <b>rmam</b>  |
|           | <b>enter_ca_mode</b>         | <b>smcup</b> |
|           | <b>clear_screen</b>          | <b>clear</b> |
|           | <b>set_attributes</b>        | <b>sgr</b>   |
| Ending:   |                              |              |
|           | <b>change_scroll_region</b>  | <b>csr</b>   |
|           | <b>cursor_address</b>        | <b>cup</b>   |
|           | <b>clr_eol</b>               | <b>el</b>    |
|           | <b>exit_ca_mode</b>          | <b>rmcup</b> |

(Continued on next page)

(Continued)

| Variable Name                     | Capname            |
|-----------------------------------|--------------------|
| <code>enter_am_mode</code>        | <code>smam</code>  |
| <code>exit_attribute_mode</code>  | <code>sgr0</code>  |
| <code>cursor_normal</code>        | <code>vnorm</code> |
| <code>exit_pc_charset_mode</code> | <code>rmpch</code> |
| <code>user2</code>                | <code>u2</code>    |
| <code>exit_scancode_mode</code>   | <code>rmsc</code>  |
| <code>user3</code>                | <code>u3</code>    |

### *SCO scancode terminal support*

SCO UNIX has an API (Application Program Interface) that provides a higher level of control over the switching of PC scancode terminals to and from PC scancode mode.

When the SCO scancode API is available, DOS Services uses it instead of switching the mode of the terminal itself. Rather than using `smsc`, `rmsc`, `u0`, `u1`, `u2`, or `u3`, DOS Services uses the API to put the terminal into scancode mode. In all other respects, DOS Services uses the terminal in scancode mode just as it would if the API did not exist.

The SCO scancode API uses `smsc` and `rmsc` to switch modes, just as DOS Services does. The API does *not* use `u0`, `u1`, or `u2` to set alternate XON/XOFF, nor does it use `u3` or the `USE_TERM` option in `pctrm`.

The API also needs some additional descriptors. Refer to the documentation on this API for more information on how it works and any special support it might need in `terminfo`.

A single `terminfo` can serve both purposes. If you are writing a `terminfo` and you want it to work in all cases, whether the API is available or not, specify all of the above in the file. Include the descriptors needed by DOS Services and those needed by the API. The appropriate items are used or bypassed, according to whether the API is present or absent.

## *Output on PC scancode terminals*

This section discusses `terminfo` items used to display DOS characters on PC terminals.

In displaying characters on the screen, the only difference between standard serial terminals and PC terminals is the character set used. PC terminals use a PC character set; standard terminals use either the ASCII or the ISO 8859 character set. Typically, the PC character set used is based on DOS code page 437.

DOS Services uses **terminfo** exactly the same way. The only difference lies in the **dispc terminfo** descriptor. This section discusses the form of **dispc** used for those terminals that can display the entire PC character set.

As previously discussed, the **dispc terminfo** descriptor is used to describe how PC characters are to be displayed on the terminal. There are two forms of usage of the **dispc** descriptor. The first is the filename of a PCT file, which contains information about how each character is to be displayed. This form *can* be used for PC terminals, but a second form is preferred.

The second form for **dispc** is a **terminfo parameterized string**; it defines a conversion from PC characters to terminal commands that display each character. This second form is preferred for several reasons. It is more efficient, self contained, and requires no external PCT file.

**dispc** has two forms because different methods work better on different terminal types. A parameterized string can be used for PC terminals because the rules for displaying PC characters are normally very easy and regular. Creating a parameterized string **dispc** for an ASCII terminal would normally be too complex for the limited size of a parameterized string.

## *Parameterized strings*

---

A parameterized string is a small logic program that takes input, processes it, and produces output. It uses a *stack* to hold temporary values, handling it in the manner of a Reverse Polish Notation (postfix) calculator.

Logically, the program is called with a character as a parameter. It converts the input character to a string of zero or more output characters. The program can do arithmetic (addition, subtraction, multiplication, division, negation, and mod). It can also do logical operations (and, or, equal to, less than, greater than), and bit operations (and, or, xor, not). The logical operations are used in an *if-then-else* form that can be fully nested.

For more information about parameterized strings, see the **terminfo(M)** manual page.

## *dispc as a parameterized string*

---

The one thing to note about using the **dispc** parameterized string is that the input character is the *second* parameter (%p2), not the first. (None of the other parameters are used internally by **dispc**.) This unusual usage is necessary because the **terminfo** library makes assumptions about the first parameters to parameterized strings that are not correct for what **dispc** needs to do.

The following example shows how to create a **dispc** parameterized string for the Kimtron KT70 PC terminal. The PC characters on the KT70 are displayed using these rules:

- For characters 00-1f and 7f (decimal 0-31,127), precede each with the ESC character.
- For characters 20-7e,80-ff (decimal 32-255, except 127), just output the character.

The parameterized string that encodes this rule is:

```
%?%p2%{32}%<%p2%{127}%=% | %t\E%p2%c%e%p2%c%;
```

This may seem confusing at first glance. Actually it is quite straightforward. The “%” character identifies operators. (For a full description of the operators see the **terminfo(M)** man page.) Here is a breakdown of the string:

| Operator    |       | Meaning                                 |
|-------------|-------|-----------------------------------------|
| %?          | means | "if"                                    |
| %p2%{32}%<  | means | "input character less than 32 decimal?" |
| %p2%{127}%= | means | "input character equal to 127 decimal?" |
| %           | means | "logical or of the previous two tests"  |
| %t          | means | "then"                                  |
| \E          | means | "output the escape character"           |
| %p2%c       | means | "output the input character"            |
| %e          | means | "else"                                  |
| %p2%c       | means | "output the input character"            |
| %;          | means | "endif"                                 |

Creating a parameterized string for **dispc** is a two-step process. First, determine the *if-then-else* logic for displaying PC characters. Then translate it into the postfix parameterized string.

## Code pages

The particular DOS code page (character set) displayed by the terminal when in PC mode is fixed. By default, it is assumed to be code page 437.

If the terminal uses another code page, put the **PCMODECP** option in the **pctrm** string. The syntax is **PCMODECP=pcnnn**, where *nnn* is the number of the code page used by the terminal. As with the other **pctrm** options, any combination and order of options is valid within the **pctrm** string. Each option is delimited by colons.

For example, if the code page is 850, then **pctrm** would be set like this:

```
pctrm=:PCMODECP=pc850;
```



## Input on PC scancode terminals

Input on PC terminals is not at all complex. They send exactly the scancodes needed by the DOS session. Thus, for normal use of the keyboard, nothing special is needed in **terminfo** except the codes that switch the terminal into and out of PC scancode mode.

Nevertheless, there are some special action escape sequences. These are used for the same functions as previously described in "Input on standard terminals"—redrawing the screen, forking a new shell, terminating or suspending DOS, shifting the screen, and displaying DOS devices.

The default lead-in is the two-key sequence, (Left-Ctrl)(Esc), that is, pressing the (Esc) key while the left (Ctrl) key is pressed. Then you release those keys. This lead-in is followed by pressing (Ctrl) and another character. The second sequence determines which function is to be done.

An alternative is to press and hold the left (Ctrl) key. While holding it down, press (Esc), let it up, and press another key.

| Press (Left-Ctrl)<br>(Esc) plus | To cause<br>this action                                                 |
|---------------------------------|-------------------------------------------------------------------------|
| <b>^R</b>                       | Redraw the screen.                                                      |
| <b>^F</b>                       | Fork a new shell.                                                       |
| <b>^K</b>                       | Kill DOS.                                                               |
| <b>^U</b>                       | See top or bottom 24 lines (when the display is not a 25-line display). |
| <b>^Z</b>                       | Suspend DOS (on systems that support suspending processes).             |
| <b>^I</b>                       | Display DOS devices.                                                    |

For example, to kill DOS, you press the left (Ctrl) key and hold it. While keeping it down, press and release the (Esc). Still holding the left (Ctrl), press and release the "K" key. You can also release and re-press the left (Ctrl) key between the (Esc) and the "K" key. Either method works.

### Changing the lead-in sequence

Using **terminfo**, the (Left-Ctrl)(Esc) lead-in for the sequence can be changed to any other key combination. The **terminfo** **scesc** descriptor is used to specify a different initial key.

| Variable Name          | Capname      |
|------------------------|--------------|
| <b>scancode_escape</b> | <b>scesc</b> |

The **scesc** string must have at least one character. The first is the value of the scancode for the key. The following characters are the scancode values of the (Shift)-type keys that must be pressed at the same time.

| Shift-type Key   | Octal Sequence |
|------------------|----------------|
| ⟨Left Ctrl⟩      | 35             |
| ⟨Right Ctrl⟩     | 340 35         |
| ⟨Left Alt⟩       | 70             |
| ⟨Right Alt⟩      | 340 70         |
| ⟨Left Shift⟩     | 52             |
| ⟨Right Shift⟩    | 66             |
| E0-⟨Left Shift⟩  | 340 52         |
| E0-⟨Right Shift⟩ | 340 66         |

For example, suppose the lead-in is to be the ⟨F12⟩ function key (scancode value octal 130), with the left ⟨Shift⟩ and the left ⟨Ctrl⟩ keys held down. For this ⟨Ctrl⟩⟨Shift⟩⟨F12⟩ lead-in, **scesc** would be set like this:

**scesc=\130\035\052**

## PC-character-to-ASCII conversions

The following table shows the built-in PC-character-to-7-bit-ASCII conversions used to display PC characters on a serial terminal. The first field is the PC character number (in hex). The second is the 7-bit ASCII character to be displayed.

|     |   |     |    |     |    |     |   |     |   |      |   |     |   |     |    |
|-----|---|-----|----|-----|----|-----|---|-----|---|------|---|-----|---|-----|----|
| 00h | * | 20h | SP | 40h | at | 60h | ` | 80h | c | A0h  | a | C0h | + | E0h | *  |
| 01h | * | 21h | !  | 41h | A  | 61h | a | 81h | u | A1h  | i | C1h | + | E1h | *  |
| 02h | * | 22h | "  | 42h | B  | 62h | b | 82h | e | A2h  | o | C2h | + | E2h | *  |
| 03h | * | 23h | #  | 43h | C  | 63h | c | 83h | a | A3h  | u | C3h | + | E3h | *  |
| 04h | * | 24h | \$ | 44h | D  | 64h | d | 84h | a | A4h  | n | C4h | - | E4h | *  |
| 05h | * | 25h | %  | 45h | E  | 65h | e | 85h | a | A5h  | N | C5h | + | E5h | *  |
| 06h | * | 26h | &  | 46h | F  | 66h | f | 86h | a | A6h  | a | C6h | + | E6h | *  |
| 07h | * | 27h | '  | 47h | G  | 67h | g | 87h | c | A7h  | o | C7h | + | E7h | *  |
| 08h | * | 28h | {  | 48h | H  | 68h | h | 88h | e | A8h  | ? | C8h | + | E8h | *  |
| 09h | * | 29h | }  | 49h | I  | 69h | i | 89h | e | A9h  | - | C9h | + | E9h | *  |
| 0Ah | * | 2Ah | *  | 4Ah | J  | 6Ah | j | 8Ah | e | AAh  | - | CAh | + | EAh | *  |
| 0Bh | * | 2Bh | +  | 4Bh | K  | 6Bh | k | 8Bh | i | ABh  | * | CBh | + | EBh | *  |
| 0Ch | * | 2Ch | ,  | 4Ch | L  | 6Ch | l | 8Ch | i | ACH  | * | CCh | + | ECh | *  |
| 0Dh | * | 2Dh | -  | 4Dh | M  | 6Dh | m | 8Dh | i | ADh  | ! | CDh | = | EDh | *  |
| 0Eh | * | 2Eh | .  | 4Eh | N  | 6Eh | n | 8Eh | A | A Eh | < | CEh | + | EEh | *  |
| 0Fh | * | 2Fh | /  | 4Fh | O  | 6Fh | o | 8Fh | A | AFh  | > | CFh | + | EFh | *  |
| 10h | > | 30h | 0  | 50h | P  | 70h | p | 90h | E | B0h  | * | D0h | + | F0h | =  |
| 11h | < | 31h | 1  | 51h | Q  | 71h | q | 91h | a | B1h  | * | D1h | + | F1h | +  |
| 12h | * | 32h | 2  | 52h | R  | 72h | r | 92h | A | B2h  | * | D2h | + | F2h | >  |
| 13h | * | 33h | 3  | 53h | S  | 73h | s | 93h | o | B3h  |   | D3h | + | F3h | <  |
| 14h | * | 34h | 4  | 54h | T  | 74h | t | 94h | o | B4h  | + | D4h | + | F4h | *  |
| 15h | * | 35h | 5  | 55h | U  | 75h | u | 95h | o | B5h  | + | D5h | + | F5h | *  |
| 16h | - | 36h | 6  | 56h | V  | 76h | v | 96h | u | B6h  | + | D6h | + | F6h | *  |
| 17h | * | 37h | 7  | 57h | W  | 77h | w | 97h | u | B7h  | + | D7h | + | F7h | ~  |
| 18h | ^ | 38h | 8  | 58h | X  | 78h | x | 98h | y | B8h  | + | D8h | + | F8h | 0  |
| 19h | v | 39h | 9  | 59h | Y  | 79h | y | 99h | o | B9h  | + | D9h | + | F9h | *  |
| 1Ah | > | 3Ah | :  | 5Ah | Z  | 7Ah | z | 9Ah | u | BAh  |   | DAh | + | FAh | *  |
| 1Bh | < | 3Bh | ;  | 5Bh | [  | 7Bh | { | 9Bh | c | BBh  | + | DBh | * | FBh | V  |
| 1Ch | * | 3Ch | <  | 5Ch | \  | 7Ch |   | 9Ch | L | BCh  | + | DCh | * | FCh | n  |
| 1Dh | * | 3Dh | =  | 5Dh | ]  | 7Dh | } | 9Dh | Y | BDh  | + | DDh | * | FDh | 2  |
| 1Eh | ^ | 3Eh | >  | 5Eh | ^  | 7Eh | ~ | 9Eh | P | BEh  | + | DEh | * | FEh | *  |
| 1Fh | v | 3Fh | ?  | 5Fh | _  | 7Fh | * | 9Fh | S | BFh  | + | DFh | * | FFh | SP |

|     |   |     |    |     |   |     |   |     |   |     |   |     |   |     |    |
|-----|---|-----|----|-----|---|-----|---|-----|---|-----|---|-----|---|-----|----|
| 00h | * | 20h | SP | 40h | @ | 60h | ' | 80h | c | a0h | a | c0h | + | e0h | *  |
| 01h | * | 21h | !  | 41h | A | 61h | a | 81h | u | a1h | i | c1h | + | e1h | *  |
| 02h | * | 22h | "  | 42h | B | 62h | b | 82h | e | a2h | o | c2h | + | e2h | *  |
| 03h | * | 23h | #  | 43h | C | 63h | c | 83h | a | a3h | u | c3h | + | e3h | *  |
| 04h | * | 24h | \$ | 44h | D | 64h | d | 84h | a | a4h | n | c4h | - | e4h | *  |
| 05h | * | 25h | %  | 45h | E | 65h | e | 85h | a | a5h | N | c5h | + | e5h | *  |
| 06h | * | 26h | &  | 46h | F | 66h | f | 86h | a | a6h | a | c6h | + | e6h | *  |
| 07h | * | 27h | '  | 47h | G | 67h | g | 87h | c | a7h | o | c7h | + | e7h | *  |
| 08h | * | 28h | (  | 48h | H | 68h | h | 88h | e | a8h | ? | c8h | + | e8h | *  |
| 09h | * | 29h | )  | 49h | I | 69h | i | 89h | e | a9h | - | c9h | + | e9h | *  |
| 0ah | * | 2ah | *  | 4ah | J | 6ah | j | 8ah | e | aah | - | cah | + | eah | *  |
| 0bh | * | 2bh | +  | 4bh | K | 6bh | k | 8bh | i | abh | * | cbh | + | ebh | *  |
| 0ch | * | 2ch | ,  | 4ch | L | 6ch | l | 8ch | i | ach | * | cch | + | ech | *  |
| 0dh | * | 2dh | -  | 4dh | M | 6dh | m | 8dh | i | adh | ! | cdh | = | edh | *  |
| 0eh | * | 2eh | .  | 4eh | N | 6eh | n | 8eh | A | ae  | < | ceh | + | eeh | *  |
| 0fh | * | 2fh | /  | 4fh | O | 6fh | o | 8fh | A | afh | > | cfh | + | efh | *  |
| 10h | > | 30h | 0  | 50h | P | 70h | p | 90h | E | b0h | * | d0h | + | f0h | =  |
| 11h | < | 31h | 1  | 51h | Q | 71h | q | 91h | a | b1h | * | d1h | + | f1h | +  |
| 12h | * | 32h | 2  | 52h | R | 72h | r | 92h | A | b2h | * | d2h | + | f2h | >  |
| 13h | * | 33h | 3  | 53h | S | 73h | s | 93h | o | b3h |   | d3h | + | f3h | <  |
| 14h | * | 34h | 4  | 54h | T | 74h | t | 94h | o | b4h | + | d4h | + | f4h | *  |
| 15h | * | 35h | 5  | 55h | U | 75h | u | 95h | o | b5h | + | d5h | + | f5h | *  |
| 16h | - | 36h | 6  | 56h | V | 76h | v | 96h | u | b6h | + | d6h | + | f6h | *  |
| 17h | * | 37h | 7  | 57h | W | 77h | w | 97h | u | b7h | + | d7h | + | f7h | ~  |
| 18h | ^ | 38h | 8  | 58h | X | 78h | x | 98h | y | b8h | + | d8h | + | f8h | O  |
| 19h | v | 39h | 9  | 59h | Y | 79h | y | 99h | o | b9h | + | d9h | + | f9h | *  |
| 1ah | > | 3ah | :  | 5ah | Z | 7ah | z | 9ah | u | bah |   | dah | + | fah | *  |
| 1bh | < | 3bh | ;  | 5bh | [ | 7bh | { | 9bh | c | b   | + | db  | * | fb  | V  |
| 1ch | * | 3ch | <  | 5ch | \ | 7ch |   | 9ch | L | b   | + | d   | * | f   | n  |
| 1dh | * | 3dh | =  | 5dh | ] | 7dh | } | 9dh | Y | b   | + | d   | * | f   | 2  |
| 1eh | ^ | 3eh | >  | 5eh | ^ | 7eh | ~ | 9eh | P | b   | + | d   | * | f   | *  |
| 1fh | v | 3fh | ?  | 5fh | _ | 7fh | * | 9fh | S | b   | + | d   | * | f   | SP |

## terminfo escape sequences

These **terminfo** escape sequences provide alternatives to the standard, built-in escape sequences for each of the functions. Both the default sequences and those specified in **terminfo** work concurrently. The ones in **terminfo** take precedence over the defaults.

| terminfo<br>Variable Name | terminfo<br>Capname | AT Keyboard<br>Key Name | Default<br>Escape Sequence |
|---------------------------|---------------------|-------------------------|----------------------------|
| <b>Numeric Keypad</b>     |                     |                         |                            |
| key_sleft                 | kLFT                | Left Arrow              | 4    Esc f                 |
| key_sright                | kRIT                | Right Arrow             | 6    Esc g                 |
| key_sr                    | kri                 | Up Arrow                | 8    Esc t                 |
| key_sf                    | kind                | Down Arrow              | 2    Esc v                 |
| key_shome                 | kHOM                | Home                    | 7    Esc h                 |
| key_previous              | kprv                | Pg Up                   | 9    Esc p                 |
| key_next                  | knxt                | Pg Down                 | 3    Esc q                 |
| key_il                    | kil1                | Ins                     | 0    Esc i                 |
| key_sdc                   | kDC                 | Del                     | .    Esc d                 |
| key_send                  | kEND                | End                     | 1    Esc e                 |
| key_b2                    | kb2                 | Center Key              | 5    Esc .                 |
| key_f53                   | kf13                | Num Lock                | Esc n                      |
| key_f54                   | kf54                | / Key                   | Esc [                      |
| key_f55                   | kf55                | * Key                   | Esc *                      |
| key_f56                   | kf56                | - Key                   | Esc -                      |
| key_f57                   | kf57                | + Key                   | Esc +                      |
| key_enter                 | kent                | Enter                   | Esc ^                      |
| <b>Shift Keys</b>         |                     |                         |                            |
| key_f58                   | kf58                | Left Shift              | Esc z                      |
| key_f59                   | kf59                | Right Shift             | Esc /                      |
| key_f60                   | kf60                | Ctrl                    | Esc c                      |
| key_f61                   | kf61                | Alt                     | Esc a                      |
| key_f62                   | kf62                | Right Ctrl              | Esc C                      |
| key_f63                   | kf63                | Right Alt               | Esc A                      |

| terminfo<br>Variable Name | terminfo<br>Capname | AT Keyboard<br>Key Name | Default<br>Escape Sequence |
|---------------------------|---------------------|-------------------------|----------------------------|
| <b>Cursor Keypad</b>      |                     |                         |                            |
| key_il                    | kill                | Insert                  | Esc I                      |
| key_home                  | kHome               | Home                    | Esc H                      |
| key_ppage                 | kpp                 | Page Up                 | Esc P                      |
| key_dc                    | kDC                 | Delete                  | Esc D                      |
| key_end                   | kEND                | End                     | Esc E                      |
| key_npage                 | knxt                | Page Down               | Esc Q                      |
| key_up                    | kri                 | Up Arrow                | Esc T                      |
| key_left                  | kLFT                | Left Arrow              | Esc F                      |
| key_down                  | kind                | Down Arrow              | Esc V                      |
| key_right                 | kRIT                | Right Arrow             | Esc G                      |
| <b>Special Keys</b>       |                     |                         |                            |
| key_print                 | kpri                | Print Screen            | Esc r                      |
| key_f50                   | kf50                | Scroll Lock             | Esc s                      |
| key_restart               | krst                | Pause/Break             | Esc                        |
| key_f51                   | kf51                | Caps Lock               | Esc k                      |
| key_f52                   | kf52                | 102nd key               | Esc <                      |
| key_options               | kopt                | Sys Req key             | Esc =                      |
| key_sic                   | kIC                 | Esc key                 | Esc esc                    |
| <b>Function Keys</b>      |                     |                         |                            |
| key_f1                    | kf1                 | F1                      | Esc 1                      |
| key_f2                    | kf2                 | F2                      | Esc 2                      |
| key_f3                    | kf3                 | F3                      | Esc 3                      |
| key_f4                    | kf4                 | F4                      | Esc 4                      |
| key_f5                    | kf5                 | F5                      | Esc 5                      |
| key_f6                    | kf6                 | F6                      | Esc 6                      |
| key_f7                    | kf7                 | F7                      | Esc 7                      |
| key_f8                    | kf8                 | F8                      | Esc 8                      |
| key_f9                    | kf9                 | F9                      | Esc 9                      |
| key_f10                   | kf10                | F10                     | Esc 0                      |
| key_f11                   | kf11                | F11                     | Esc ;                      |
| key_f12                   | kf12                | F12                     | Esc :                      |

| terminfo<br>Variable Name       | terminfo<br>Capname | AT Keyboard<br>Key Name | Default<br>Escape Sequence |
|---------------------------------|---------------------|-------------------------|----------------------------|
| <b>Shifted Function Keys</b>    |                     |                         |                            |
| key_f13                         | kf13                | Shift F1                |                            |
| key_f14                         | kf14                | Shift F2                |                            |
| key_f15                         | kf15                | Shift F3                |                            |
| key_f16                         | kf16                | Shift F4                |                            |
| key_f16                         | kf17                | Shift F5                |                            |
| key_f18                         | kf18                | Shift F6                |                            |
| key_f19                         | kf19                | Shift F7                |                            |
| key_f20                         | kf20                | Shift F8                |                            |
| key_f21                         | kf21                | Shift F9                |                            |
| key_f22                         | kf22                | Shift F10               |                            |
| key_f23                         | kf23                | Shift F11               |                            |
| key_f24                         | kf24                | Shift F12               |                            |
| <b>Controlled Function Keys</b> |                     |                         |                            |
| key_f25                         | kf25                | Ctrl F1                 |                            |
| key_f26                         | kf26                | Ctrl F2                 |                            |
| key_f27                         | kf27                | Ctrl F3                 |                            |
| key_f28                         | kf28                | Ctrl F4                 |                            |
| key_f29                         | kf29                | Ctrl F5                 |                            |
| key_f30                         | kf30                | Ctrl F6                 |                            |
| key_f31                         | kf31                | Ctrl F7                 |                            |
| key_f32                         | kf32                | Ctrl F8                 |                            |
| key_f33                         | kf33                | Ctrl F9                 |                            |
| key_f34                         | kf34                | Ctrl F10                |                            |
| key_f35                         | kf35                | Ctrl F11                |                            |
| key_f36                         | kf36                | Ctrl F12                |                            |

| terminfo<br>Variable Name  | terminfo<br>Capname | AT Keyboard<br>Key Name | Default<br>Escape Sequence |
|----------------------------|---------------------|-------------------------|----------------------------|
| <b>Alted Function Keys</b> |                     |                         |                            |
| key_f37                    | kf37                | Alt F1                  |                            |
| key_f38                    | kf38                | Alt F2                  |                            |
| key_f39                    | kf39                | Alt F3                  |                            |
| key_f40                    | kf40                | Alt F4                  |                            |
| key_f41                    | kf41                | Alt F5                  |                            |
| key_f42                    | kf42                | Alt F6                  |                            |
| key_f43                    | kf43                | Alt F7                  |                            |
| key_f44                    | kf44                | Alt F8                  |                            |
| key_f45                    | kf45                | Alt F9                  |                            |
| key_f46                    | kf46                | Alt F10                 |                            |
| key_f47                    | kf47                | Alt F11                 |                            |
| key_f48                    | kf48                | Alt F12                 |                            |
| <b>Special Actions</b>     |                     |                         |                            |
| key_f0                     | kf0                 | All Shifts Released     | Esc Space                  |
| key_beg                    | kbeg                | TopBottom Shift Screen  | Esc ^U                     |
| key_resume                 | kres                | Fork A New Shell        | Esc ^F                     |
| key_refresh                | krfr                | Redraw Screen           | Esc ^R                     |
| key_exit                   | kext                | Kill DOS                | Esc ^K                     |
| key_suspend                | kspd                | Suspend DOS             | Esc ^Z                     |
| key_reference              | kref                | Display DOS Devices     | Esc ^I                     |



# *Administering Networking Services*

|                                                    |     |
|----------------------------------------------------|-----|
| <b>Introduction to networking</b> .....            | 402 |
| <b>How the network works</b> .....                 | 403 |
| OSI .....                                          | 404 |
| TCP/IP .....                                       | 405 |
| Correspondence of models .....                     | 406 |
| Designing or adding to your network .....          | 406 |
| Common networking administration tasks .....       | 406 |
| <b>Networking services</b> .....                   | 407 |
| <b>Introducing TCP/IP</b> .....                    | 408 |
| The internet protocol (IP) .....                   | 408 |
| The transmission control protocol (TCP) .....      | 409 |
| Other TCP/IP protocols .....                       | 409 |
| TCP/IP end-user commands .....                     | 410 |
| <b>Configuring TCP/IP</b> .....                    | 410 |
| System name .....                                  | 411 |
| Driver type .....                                  | 411 |
| Interrupt vector .....                             | 411 |
| I/O base address .....                             | 411 |
| Thick/thin cable .....                             | 412 |
| RAM buffer size and base address .....             | 412 |
| Domain name .....                                  | 412 |
| IP address .....                                   | 413 |
| Broadcast address parameters .....                 | 414 |
| netmask setting .....                              | 415 |
| <b>TCP/IP over a SLIP or PPP serial line</b> ..... | 415 |
| tty line .....                                     | 415 |
| Source IP address .....                            | 416 |
| Destination IP address .....                       | 416 |
| Baud rate .....                                    | 416 |
| SLIP netmask .....                                 | 416 |
| <b>Maintaining TCP/IP</b> .....                    | 416 |
| Adding hosts .....                                 | 416 |
| Configuring the name domain server .....           | 416 |
| Setting up routing tables .....                    | 416 |
| Establishing user equivalence .....                | 417 |
| Setting up anonymous ftp .....                     | 417 |
| Altering installation parameters .....             | 417 |
| Tuning kernel parameters .....                     | 417 |

|                                             |            |
|---------------------------------------------|------------|
| Monitoring TCP/IP status .....              | 417        |
| Enabling remote printing .....              | 417        |
| <b>Introducing NFS .....</b>                | <b>418</b> |
| Maintaining NFS .....                       | 418        |
| <b>Introducing LAN Manager client .....</b> | <b>419</b> |
| Configuring LAN Manager Client .....        | 420        |
| Maintaining LAN Manager Client .....        | 420        |
| <b>Introducing UUCP .....</b>               | <b>421</b> |
| Configuring UUCP .....                      | 421        |
| <b>Introducing MMDF .....</b>               | <b>422</b> |
| Configuring MMDF .....                      | 422        |
| <b>Other networking software .....</b>      | <b>422</b> |

## *Chapter 2*

# *TCP/IP network administration* **423**

---

|                                                 |            |
|-------------------------------------------------|------------|
| <b>Kernel configuration .....</b>               | <b>423</b> |
| <b>Setting interface parameters .....</b>       | <b>427</b> |
| <b>Creating a subnetwork .....</b>              | <b>427</b> |
| <b>Network servers .....</b>                    | <b>428</b> |
| <b>Network databases .....</b>                  | <b>429</b> |
| Establishing user equivalence .....             | 429        |
| Setting up anonymous ftp .....                  | 430        |
| <b>Administering pseudo ttys .....</b>          | <b>432</b> |
| <b>Network tuning and troubleshooting .....</b> | <b>433</b> |
| STREAMS tuning .....                            | 433        |
| Active connections display .....                | 435        |
| netstat -a .....                                | 435        |
| Descriptions of the display headings .....      | 436        |
| Interfaces .....                                | 436        |
| netstat -i .....                                | 436        |
| Descriptions of the display headings .....      | 436        |
| Routing tables .....                            | 437        |
| netstat -r .....                                | 437        |
| Descriptions of the display headings .....      | 438        |
| Statistics display .....                        | 438        |
| netstat -s .....                                | 439        |

## Chapter 3

# TCP/IP serial line communications

441

|                                                              |     |
|--------------------------------------------------------------|-----|
| <b>Administering SLIP</b> .....                              | 441 |
| Configuring a SLIP connection .....                          | 442 |
| Preparing to configure a SLIP connection .....               | 442 |
| Configuring a direct SLIP connection .....                   | 442 |
| Configuring a dialup SLIP connection .....                   | 444 |
| Configuring a SLIP/Ethernet or SLIP/Token-Ring gateway ..... | 446 |
| Removing SLIP .....                                          | 447 |
| Troubleshooting SLIP configurations .....                    | 447 |
| Common problems with SLIP .....                              | 447 |
| Verifying serial cable connectivity .....                    | 448 |
| Troubleshooting problems with ping .....                     | 448 |
| Troubleshooting problems with rlogin or telnet .....         | 449 |
| More SLIP information .....                                  | 450 |
| <b>Administering PPP</b> .....                               | 450 |
| Interoperability with other PPP implementations .....        | 451 |
| PPP compared to SLIP .....                                   | 453 |
| Configuring PPP .....                                        | 454 |
| Preparing to configure PPP .....                             | 454 |
| Configuring PPP with netconfig .....                         | 454 |
| Adding PPP information to configuration files .....          | 457 |
| Configuring a PPP/Ethernet or PPP/Token-Ring gateway .....   | 458 |
| Removing PPP .....                                           | 458 |
| Troubleshooting PPP .....                                    | 458 |
| More PPP information .....                                   | 460 |

## Chapter 4

# Configuring the BIND name server

461

|                               |     |
|-------------------------------|-----|
| <b>The name service</b> ..... | 461 |
| <b>Types of servers</b> ..... | 462 |
| Master servers .....          | 462 |
| Primary .....                 | 462 |
| Secondary .....               | 462 |
| Caching-only servers .....    | 463 |
| Remote servers .....          | 463 |
| Slave server .....            | 463 |

|                                                 |     |
|-------------------------------------------------|-----|
| <b>Setting up your own domain</b>               | 464 |
| Internet                                        | 464 |
| BITNET                                          | 464 |
| Boot file                                       | 464 |
| Directory                                       | 464 |
| Primary master                                  | 465 |
| Secondary master                                | 465 |
| Caching-only server                             | 465 |
| Forwarders                                      | 466 |
| Slave mode                                      | 466 |
| <b>Remote servers</b>                           | 466 |
| <b>Initializing the cache</b>                   | 466 |
| <b>Standard files</b>                           | 467 |
| <b>Standard resource records</b>                | 467 |
| Separating data into multiple files             | 468 |
| Changing an origin in a data file               | 469 |
| The start of authority resource record (SOA)    | 469 |
| The name server resource record (NS)            | 470 |
| The address resource record (A)                 | 470 |
| The host information resource record (HINFO)    | 470 |
| The well-known services resource record (WKS)   | 471 |
| The canonical name resource record (CNAME)      | 471 |
| The domain name pointer resource record (PTR)   | 471 |
| The mailbox resource record (MB)                | 472 |
| The mail rename resource record (MR)            | 472 |
| The mailbox information resource record (MINFO) | 472 |
| The mail group member resource record (MG)      | 472 |
| The mail exchanger resource record (MX)         | 473 |
| <b>Some sample files</b>                        | 473 |
| Caching-only server                             | 474 |
| Primary master server                           | 474 |
| Secondary master server                         | 474 |
| The /etc/resolv.conf file                       | 474 |
| root.cache                                      | 475 |
| named.local                                     | 475 |
| named.hosts                                     | 476 |
| named.rev                                       | 477 |
| <b>Additional sample files</b>                  | 477 |
| named.boot                                      | 477 |
| root.cache                                      | 478 |
| named.local                                     | 478 |

|                                |            |
|--------------------------------|------------|
| mynet-host.s.rev .....         | 479        |
| mynet.soa .....                | 479        |
| <b>Domain management .....</b> | <b>479</b> |
| Starting the name server ..... | 479        |
| /etc/named.pid .....           | 480        |
| /etc/hosts .....               | 480        |
| Reload .....                   | 480        |
| Debugging .....                | 480        |
| More BIND information .....    | 481        |

## Chapter 5

# ***Gateways and routing*** **483**

|                                 |     |
|---------------------------------|-----|
| Running routed .....            | 484 |
| Running gated .....             | 485 |
| Sample configuration file ..... | 486 |
| More gated information .....    | 489 |

## Chapter 6

# ***Configuring and using SNMP*** **491**

|                                                     |            |
|-----------------------------------------------------|------------|
| <b>Basic concepts .....</b>                         | <b>491</b> |
| The SNMP protocol .....                             | 491        |
| SMI: Structure of Management Information .....      | 492        |
| MIB: the Management Information Base .....          | 494        |
| SMUX (SNMP Multiplexing) protocol information ..... | 494        |
| <b>Other concepts .....</b>                         | <b>495</b> |
| Agents and management stations .....                | 495        |
| Traps .....                                         | 495        |
| Authentication .....                                | 495        |
| <b>Overview of the SCO implementation .....</b>     | <b>496</b> |
| <b>Configuring the SNMP agent .....</b>             | <b>497</b> |
| <b>Using the SNMP commands .....</b>                | <b>499</b> |
| <b>Using SNMP to correct problems .....</b>         | <b>504</b> |
| Obtaining remote system contacts .....              | 504        |
| Removing an incorrect routing entry .....           | 504        |
| Marking an interface down .....                     | 505        |
| Removing an incorrect ARP entry .....               | 506        |
| <b>More SNMP information .....</b>                  | <b>506</b> |

## Chapter 7

# Remote line printing

507

---

|                                   |     |
|-----------------------------------|-----|
| Installing and removing RLP ..... | 508 |
| How RLP works .....               | 509 |
| Using RLP .....                   | 510 |
| SCO clients .....                 | 510 |
| 4.3BSD clients .....              | 511 |
| Setting up a client .....         | 511 |
| Setting up a print server .....   | 514 |
| Deleting printcap entries .....   | 515 |

## Chapter 8

# Synchronizing clocks

517

---

|                                             |     |
|---------------------------------------------|-----|
| Time synchronization protocol .....         | 517 |
| How the time daemon works .....             | 518 |
| Guidelines .....                            | 519 |
| Options .....                               | 520 |
| Daily operation .....                       | 520 |
| Network time protocol .....                 | 521 |
| Important terms .....                       | 521 |
| Overview .....                              | 524 |
| Guidelines .....                            | 524 |
| An example synchronization subnet .....     | 525 |
| The NTP configuration file .....            | 526 |
| Configuration statements .....              | 527 |
| Example ntp.conf file .....                 | 530 |
| The keys file .....                         | 530 |
| The clock.txt file .....                    | 532 |
| The driftfile .....                         | 533 |
| Association modes .....                     | 533 |
| Address and mask facility .....             | 534 |
| Name resolution .....                       | 536 |
| Sample scenarios .....                      | 537 |
| Testing and tuning .....                    | 540 |
| Query commands .....                        | 541 |
| Further examples .....                      | 542 |
| Troubleshooting .....                       | 545 |
| Running mixed synchronization subnets ..... | 546 |

---

|                                                                                    |     |
|------------------------------------------------------------------------------------|-----|
| <b>Basic concepts</b> .....                                                        | 547 |
| How files interact .....                                                           | 547 |
| How daemons interact .....                                                         | 548 |
| Managing user accounts .....                                                       | 548 |
| <b>Configuring NFS</b> .....                                                       | 549 |
| Setting up an NFS server .....                                                     | 549 |
| Enabling export .....                                                              | 549 |
| Checking the /etc/hosts file .....                                                 | 550 |
| Setting up an NFS client .....                                                     | 551 |
| Mounting the remote filesystem .....                                               | 551 |
| Enabling default mounting .....                                                    | 551 |
| Enabling user mounting .....                                                       | 553 |
| Starting and stopping NFS .....                                                    | 554 |
| <b>Administering NFS</b> .....                                                     | 554 |
| Debugging NFS .....                                                                | 554 |
| General hints .....                                                                | 555 |
| Remote mount failed .....                                                          | 556 |
| Programs are hung .....                                                            | 560 |
| Everything works slowly .....                                                      | 561 |
| Serial number validation errors .....                                              | 561 |
| Incompatibilities with remote filesystems .....                                    | 562 |
| File operations not supported .....                                                | 562 |
| Cannot access remote devices .....                                                 | 562 |
| Cannot access different filesystem types .....                                     | 562 |
| Cannot access indirect filesystems .....                                           | 562 |
| Only 8 groups per user supported when mounting a Release<br>1.1.1 filesystem ..... | 562 |
| Handling clock skew in user programs .....                                         | 563 |

---



---

|                                |     |
|--------------------------------|-----|
| <b>Understanding NIS</b> ..... | 565 |
| NIS maps .....                 | 566 |
| NIS domain .....               | 567 |
| NIS servers .....              | 567 |

---



|                                                         |            |
|---------------------------------------------------------|------------|
| NIS daemons .....                                       | 568        |
| NIS utilities .....                                     | 569        |
| NIS user accounts .....                                 | 569        |
| Understanding YP_MAP_X_LATE .....                       | 570        |
| Security limitations on configuration .....             | 570        |
| <b>Planning an NIS configuration .....</b>              | <b>571</b> |
| Selecting domains .....                                 | 571        |
| Selecting servers .....                                 | 572        |
| Security .....                                          | 572        |
| <b>Configuring NIS .....</b>                            | <b>573</b> |
| Setting up an NIS server .....                          | 573        |
| Initializing NIS .....                                  | 574        |
| Starting NIS .....                                      | 581        |
| <b>Managing NIS maps .....</b>                          | <b>582</b> |
| Map manipulation utilities .....                        | 582        |
| Modifying existing maps .....                           | 583        |
| Propagating an NIS map .....                            | 584        |
| Using cron .....                                        | 584        |
| Using the interactive approach .....                    | 585        |
| Using yppush on the master server .....                 | 585        |
| Making new NIS maps .....                               | 585        |
| Creating a new map from an ASCII file .....             | 586        |
| Creating a new map from standard input .....            | 587        |
| Propagating a new NIS map .....                         | 587        |
| <b>Managing user accounts .....</b>                     | <b>588</b> |
| NIS account types .....                                 | 588        |
| Administration utilities .....                          | 588        |
| Creating NIS accounts .....                             | 589        |
| Creating a distributed account .....                    | 589        |
| Creating a local account .....                          | 590        |
| Modifying NIS accounts .....                            | 590        |
| Changing a distributed account to a local account ..... | 590        |
| Changing a local account to a distributed account ..... | 591        |
| Modifying password file entries .....                   | 591        |
| Removing password file entries .....                    | 591        |
| <b>Administering an NIS network .....</b>               | <b>592</b> |
| Changing server configuration .....                     | 592        |
| Adding NIS servers .....                                | 592        |
| Removing servers .....                                  | 593        |
| Changing the master server .....                        | 594        |
| Debugging NIS .....                                     | 596        |

|                                                   |     |
|---------------------------------------------------|-----|
| Checking TCP/IP .....                             | 596 |
| Checking the protected password file .....        | 596 |
| Different versions of an NIS map .....            | 596 |
| ypserv crashes .....                              | 597 |
| Commands hang .....                               | 598 |
| NIS unavailable .....                             | 599 |
| ypbind crashes .....                              | 600 |
| NIS interoperability .....                        | 601 |
| Security under NIS .....                          | 602 |
| Map integration restrictions in Secure Mode ..... | 602 |
| Special NIS password change .....                 | 602 |
| Manual pages covering security .....              | 602 |
| Directory of NIS services .....                   | 603 |

## Chapter 11

# Automounter

607

|                                             |     |
|---------------------------------------------|-----|
| <b>Understanding Automounter</b> .....      | 607 |
| How Automounter works .....                 | 607 |
| Automounter maps .....                      | 611 |
| Planning an Automounter configuration ..... | 612 |
| <b>Configuring Automounter</b> .....        | 613 |
| Preparing the maps .....                    | 613 |
| Writing a master map .....                  | 613 |
| Mount point "/-" .....                      | 614 |
| Mount point /home .....                     | 614 |
| Mount point /net .....                      | 614 |
| Writing an indirect map .....               | 616 |
| Writing a direct map .....                  | 618 |
| Multiple mounts .....                       | 618 |
| Multiple locations .....                    | 620 |
| Specifying subdirectories .....             | 621 |
| Metacharacters .....                        | 622 |
| Ampersand "&" .....                         | 622 |
| Asterisk "*" .....                          | 623 |
| Backslash "\" .....                         | 624 |
| Double quotes (") .....                     | 624 |
| Environment variables .....                 | 624 |
| Including other maps .....                  | 625 |
| Starting Automounter .....                  | 625 |

|                                                      |            |
|------------------------------------------------------|------------|
| <b>Administering Automounter .....</b>               | <b>627</b> |
| Restarting Automounter .....                         | 627        |
| Modifying the maps .....                             | 627        |
| Modifying the master map .....                       | 627        |
| Modifying indirect maps .....                        | 627        |
| Modifying direct maps .....                          | 628        |
| Mount point conflicts .....                          | 628        |
| Unexpected filesystem behavior .....                 | 629        |
| Increasing mount table entries .....                 | 629        |
| Debugging Automounter .....                          | 630        |
| General error messages .....                         | 630        |
| Error messages generated by the verbose option ..... | 633        |

## **Chapter 12**

# ***LAN Manager client administration* 635**

---

|                                                                |            |
|----------------------------------------------------------------|------------|
| <b>Understanding NetBIOS .....</b>                             | <b>635</b> |
| Configuring NetBIOS to run with TCP/IP .....                   | 636        |
| Host files .....                                               | 636        |
| Broadcasts .....                                               | 636        |
| Setting the scope identifier .....                             | 638        |
| NetBIOS keepalives .....                                       | 638        |
| user ID consistency .....                                      | 639        |
| Planning for the virtual terminal program .....                | 639        |
| Accessing a server with user-level security .....              | 639        |
| <b>Adding and removing a client .....</b>                      | <b>640</b> |
| <b>Configuring the client for user access to servers .....</b> | <b>640</b> |
| Client operating parameters .....                              | 641        |
| <b>Starting and stopping the client .....</b>                  | <b>641</b> |
| <b>Managing the client .....</b>                               | <b>642</b> |
| Client status .....                                            | 642        |
| Checking NetBIOS status .....                                  | 643        |
| <b>Utility administration on the client .....</b>              | <b>645</b> |
| Configuring the client for vtp .....                           | 645        |
| <b>Modifying client parameters .....</b>                       | <b>646</b> |
| Managing NetBIOS sessions .....                                | 646        |
| Modifying the client for performance: constable .....          | 647        |
| Adjusting client read and write windows .....                  | 647        |
| Adjusting network buffer sizes .....                           | 650        |
| Parameters affecting connections to servers: lmxrc .....       | 651        |

|                                          |     |
|------------------------------------------|-----|
| Modifying kernel parameters: mtune ..... | 652 |
| TCP/IP NetBIOS driver parameters .....   | 658 |

## Chapter 13

# ***Building a remote network with UUCP*** **659**

|                                                                |     |
|----------------------------------------------------------------|-----|
| <b>What is UUCP?</b> .....                                     | 659 |
| <b>How to use this chapter</b> .....                           | 660 |
| <b>What you need</b> .....                                     | 661 |
| <b>UUCP commands</b> .....                                     | 661 |
| User programs .....                                            | 661 |
| Administrative programs .....                                  | 661 |
| UUCP directories .....                                         | 662 |
| UUCP background programs .....                                 | 662 |
| How UUCP works .....                                           | 663 |
| A sample UUCP transaction .....                                | 664 |
| <b>Configuring UUCP on your system</b> .....                   | 665 |
| Connecting a UUCP modem .....                                  | 665 |
| Variable rate modems .....                                     | 666 |
| Editing the UUCP control files .....                           | 666 |
| Creating login accounts for sites dialing-in .....             | 667 |
| UUCP anonymous login accounts .....                            | 668 |
| Testing the UUCP connection .....                              | 668 |
| <b>Detailed descriptions of UUCP configuration files</b> ..... | 669 |
| Adding entries for remote sites to the Systems file .....      | 670 |
| The schedule field .....                                       | 671 |
| The device field .....                                         | 672 |
| The speed field .....                                          | 672 |
| The phone field .....                                          | 673 |
| The login-script field .....                                   | 673 |
| Creating login scripts .....                                   | 674 |
| Setting up polling .....                                       | 676 |
| Limiting access with the Permissions file .....                | 676 |
| Structuring Permissions file entries .....                     | 676 |
| Permissions file restrictions .....                            | 677 |
| Permissions options .....                                      | 677 |
| Adding dial-out entries to the Devices file .....              | 683 |
| The type field .....                                           | 684 |
| The speed field .....                                          | 685 |
| The dialer-token field .....                                   | 685 |

|                                                                  |            |
|------------------------------------------------------------------|------------|
| Structuring dialer-token entries .....                           | 686        |
| <b>Special UUCP configuration options .....</b>                  | <b>687</b> |
| Defining a communications protocol .....                         | 688        |
| Adding dialers to the Dialers file .....                         | 688        |
| Using Dialcodes to create a portable Systems file .....          | 688        |
| Creating alternate control files with Sysfiles .....             | 688        |
| Changing uucico packet parameters .....                          | 689        |
| Preventing unknown sites from logging in .....                   | 689        |
| Configuring UUCP over TCP/IP .....                               | 689        |
| Configuring UUCP over TCP/IP with the TCP socket interface ..... | 690        |
| Configuring UUCP over TCP/IP with TLI .....                      | 692        |
| The nlsadmin(ADM) command .....                                  | 695        |
| Connecting two local systems using a direct wire .....           | 695        |
| Choosing a serial port .....                                     | 695        |
| Connecting a serial cable .....                                  | 696        |
| Edit the UUCP configuration files .....                          | 696        |
| Testing a connection .....                                       | 696        |
| <b>Complete UUCP examples .....</b>                              | <b>697</b> |
| Example 1: system gomer .....                                    | 697        |
| /etc/passwd .....                                                | 697        |
| /etc/group .....                                                 | 697        |
| /etc/systemid .....                                              | 698        |
| /etc/inittab .....                                               | 698        |
| /usr/lib/uucp/Devices .....                                      | 698        |
| /usr/lib/uucp/Permissions .....                                  | 698        |
| /usr/lib/uucp/Systems .....                                      | 699        |
| Example 2: system dingbat .....                                  | 699        |
| /etc/passwd .....                                                | 699        |
| /etc/group .....                                                 | 699        |
| /etc/systemid .....                                              | 699        |
| /etc/inittab .....                                               | 700        |
| /usr/lib/uucp/Devices .....                                      | 700        |
| /usr/lib/uucp/Permissions .....                                  | 700        |
| /usr/lib/uucp/Systems .....                                      | 700        |
| Sample commands .....                                            | 701        |
| <b>Administering your UUCP system .....</b>                      | <b>701</b> |
| UUCP maintenance shellscripts .....                              | 701        |
| Generating log reports on usage: uulog .....                     | 702        |
| uulog options .....                                              | 702        |
| Special uulog files .....                                        | 702        |
| The UUCP spool directory .....                                   | 702        |

|                                  |     |
|----------------------------------|-----|
| <b>UUCP error messages</b> ..... | 704 |
| ASSERT error messages .....      | 705 |
| UUCP STATUS error messages ..... | 707 |

## Chapter 14

# **Setting up electronic mail** **711**

|                                                               |     |
|---------------------------------------------------------------|-----|
| <b>Chapter overview</b> .....                                 | 712 |
| <b>How MMDF works</b> .....                                   | 712 |
| Outgoing mail .....                                           | 713 |
| Mail headers .....                                            | 713 |
| Incoming mail .....                                           | 716 |
| <b>The configuration utility</b> .....                        | 717 |
| Configuring communications channels .....                     | 718 |
| Configuration utility limitations .....                       | 718 |
| <b>MMDF configuration checklist</b> .....                     | 719 |
| Host name .....                                               | 721 |
| Domain name .....                                             | 721 |
| Fully qualified host name .....                               | 722 |
| Registering domain names .....                                | 723 |
| Hide your host name? .....                                    | 724 |
| Redirect mail for root? .....                                 | 724 |
| Redirect mail for mmdf? .....                                 | 725 |
| Postmaster address .....                                      | 725 |
| Setting up MMDF for UUCP .....                                | 725 |
| UUCP host name .....                                          | 725 |
| UUCP hosts with full domain names .....                       | 726 |
| Configuring MMDF to use a name server .....                   | 726 |
| Configuring smart hosts .....                                 | 726 |
| Badhosts channel .....                                        | 726 |
| Badusers channel .....                                        | 727 |
| <b>Running the configuration utility</b> .....                | 727 |
| Running the configuration utility with TCP/IP installed ..... | 734 |
| Preparing MMDF for use .....                                  | 736 |
| Restarting the deliver daemon .....                           | 736 |
| Modifying MMDF system startup .....                           | 737 |
| Other common configuration tasks .....                        | 737 |
| Creating aliases for users .....                              | 738 |
| Converting XENIX alias files .....                            | 738 |
| Changing logging levels .....                                 | 738 |

|                                           |            |
|-------------------------------------------|------------|
| Rebuilding the MMDF hashed database ..... | 738        |
| <b>MMDF configuration files .....</b>     | <b>739</b> |
| The mmdftailor file .....                 | 739        |
| MLDOMAIN .....                            | 742        |
| MLNAME and MLOCMACHINE .....              | 742        |
| UUname .....                              | 742        |
| MSUPPORT .....                            | 742        |
| Alias configuration .....                 | 743        |
| Local mail configuration .....            | 743        |
| List processing configuration .....       | 744        |
| SMTP configuration .....                  | 744        |
| Local domain table configuration .....    | 744        |
| UUCP configuration .....                  | 745        |
| The badhosts channel .....                | 745        |
| The badusers channel .....                | 745        |
| The root domain table .....               | 746        |
| Logging levels .....                      | 746        |
| Mailbox locking style .....               | 746        |
| Alias files .....                         | 747        |
| The alias.list file .....                 | 748        |
| The alias.user file .....                 | 749        |
| The alias.ali file .....                  | 749        |
| How MMDF uses alias files .....           | 750        |
| Domain files .....                        | 750        |
| Domain file format .....                  | 751        |
| The local.dom file .....                  | 751        |
| The domain.dom file .....                 | 751        |
| The uucp.dom file .....                   | 752        |
| The root.dom file .....                   | 752        |
| LAN considerations .....                  | 752        |
| Channel files .....                       | 753        |
| The local.chn file .....                  | 754        |
| The list.chn file .....                   | 754        |
| The uucp.chn file .....                   | 754        |
| The smtp.chn file .....                   | 755        |
| <b>How MMDF routes mail .....</b>         | <b>755</b> |
| Searching MMDF domain tables .....        | 756        |
| Delivery channel programs .....           | 757        |
| <b>Modifying MMDF configuration .....</b> | <b>759</b> |
| Guidelines for manual configuration ..... | 760        |
| Changing the postmaster alias .....       | 760        |

|                                                       |            |
|-------------------------------------------------------|------------|
| Maintaining user aliases .....                        | 760        |
| Changing the host name .....                          | 761        |
| Hiding your host name .....                           | 762        |
| Changing your smart host .....                        | 762        |
| Changing error logging levels .....                   | 763        |
| Converting from XENIX .....                           | 763        |
| Modifying UUCP host configuration .....               | 766        |
| Specifying options to uux .....                       | 766        |
| Adding or removing a UUCP host .....                  | 766        |
| Adding or removing an SMTP host .....                 | 767        |
| Running multiple deliver daemons .....                | 767        |
| Specifying different time intervals for deliver ..... | 768        |
| Rebuilding the MMDF hashed database .....             | 769        |
| <b>Testing MMDF configuration .....</b>               | <b>769</b> |
| Checking for MMDF problems .....                      | 769        |
| Testing mail addresses .....                          | 770        |
| <b>Maintaining the MMDF system .....</b>              | <b>770</b> |
| Checking the status of mail queues .....              | 770        |
| Removing old mail from the queues .....               | 771        |
| Monitoring the size of log files .....                | 771        |
| <b>Advanced MMDF configuration tasks .....</b>        | <b>771</b> |
| Changing the location of system mailboxes .....       | 772        |
| Specifying the MMDF "signature" .....                 | 772        |
| Configuring MMDF authorization .....                  | 772        |
| Setting authorization for hosts .....                 | 773        |
| Routing-based authorization .....                     | 775        |
| Setting authorization for users .....                 | 777        |
| Setting host and user authorization controls .....    | 779        |
| Understanding authorization log files .....           | 779        |
| Adding a new alias file .....                         | 781        |
| Adding a new domain file .....                        | 783        |
| Adding a third-party channel program .....            | 783        |
| Changing MMDF parameters .....                        | 784        |
| <b>Getting more information .....</b>                 | <b>786</b> |

## Chapter 15

# ***sendmail administration***

**789**

|                                           |            |
|-------------------------------------------|------------|
| <b>sendmail and other mailers .....</b>   | <b>790</b> |
| Comparing sendmail with delivermail ..... | 790        |



|                                                        |            |
|--------------------------------------------------------|------------|
| Comparing sendmail with MMDF .....                     | 790        |
| Sendmail and the message-processing module (MPM) ..... | 791        |
| <b>How sendmail works .....</b>                        | <b>792</b> |
| Collecting messages .....                              | 792        |
| Delivering messages .....                              | 793        |
| Queueing for retransmission .....                      | 793        |
| Return to sender .....                                 | 793        |
| Editing the message header .....                       | 794        |
| Aliasing, forwarding and including mail .....          | 794        |
| Aliasing .....                                         | 794        |
| Forwarding .....                                       | 794        |
| Including .....                                        | 794        |
| Queued messages .....                                  | 795        |
| <b>Configuring sendmail .....</b>                      | <b>795</b> |
| Configuring a standard installation .....              | 795        |
| Configuring a non-standard installation .....          | 796        |
| The syntax .....                                       | 797        |
| R and S - rewriting rules .....                        | 797        |
| D - define macro .....                                 | 797        |
| C and F - define classes .....                         | 797        |
| M - define mailer .....                                | 798        |
| H - define header .....                                | 798        |
| O - set option .....                                   | 798        |
| T - define trusted users .....                         | 799        |
| P - precedence definitions .....                       | 799        |
| The semantics .....                                    | 799        |
| Special macros, conditionals .....                     | 799        |
| Special classes .....                                  | 801        |
| The left-hand side .....                               | 801        |
| The right-hand side .....                              | 802        |
| Semantics of rewriting rule sets .....                 | 803        |
| Mailer flags .....                                     | 803        |
| The "error" mailer .....                               | 804        |
| Building a configuration file from scratch .....       | 804        |
| Purpose of the configuration table .....               | 804        |
| Relevant issues .....                                  | 804        |
| How to proceed .....                                   | 805        |
| Testing the rewriting rules: the -bt flag .....        | 805        |
| Building mailer descriptions .....                     | 806        |
| Configuration options .....                            | 808        |
| <b>Running sendmail .....</b>                          | <b>810</b> |

|                                              |     |
|----------------------------------------------|-----|
| Command line flags .....                     | 810 |
| Mailer flags .....                           | 811 |
| Arguments .....                              | 813 |
| Queue interval .....                         | 813 |
| Daemon mode .....                            | 813 |
| Forcing the queue .....                      | 813 |
| Debugging .....                              | 813 |
| Trying a different configuration file .....  | 814 |
| Changing the values of options .....         | 814 |
| Tuning .....                                 | 814 |
| Timeouts .....                               | 814 |
| Queue interval .....                         | 815 |
| Read timeouts .....                          | 815 |
| Message timeouts .....                       | 815 |
| Forking during queue runs .....              | 815 |
| Queue priorities .....                       | 815 |
| Delivery mode .....                          | 816 |
| File modes .....                             | 816 |
| To suid or not to suid? .....                | 816 |
| Temporary file modes .....                   | 817 |
| Should the alias database be writable? ..... | 817 |
| Administering sendmail .....                 | 817 |
| System log .....                             | 817 |
| The mail queue .....                         | 818 |
| Format of sendmail queue files .....         | 818 |
| Forcing the queue .....                      | 819 |
| sendmail configuration file .....            | 820 |
| The alias database .....                     | 820 |
| Rebuilding the alias database .....          | 821 |
| Potential alias database problems .....      | 821 |
| List owners .....                            | 821 |
| Per-user forwarding (.forward files) .....   | 822 |
| Special header lines .....                   | 822 |
| Return-receipt-to: .....                     | 822 |
| Errors-to: .....                             | 822 |
| Apparently-to: .....                         | 822 |
| Summary of support files .....               | 822 |
| More sendmail information .....              | 823 |

## Chapter 16

# Using modems

825

---

|                                                    |     |
|----------------------------------------------------|-----|
| Choosing a serial port .....                       | 825 |
| Configuring your modem .....                       | 826 |
| Smartmodem 1200 or compatible .....                | 826 |
| Smartmodem 2400, V-series 9600 or compatible ..... | 827 |
| Telebit Trailblazer .....                          | 828 |
| General modem configuration guidelines .....       | 828 |
| Installing a modem .....                           | 828 |
| Dialing into your computer .....                   | 834 |
| Adding passwords for dial-in lines .....           | 834 |
| Dialing out from your computer .....               | 835 |
| Using dialer programs .....                        | 835 |
| Dialers file entries .....                         | 835 |
| Dialer binaries .....                              | 836 |
| Modem commands and setup strings .....             | 836 |
| Creating a new dial dialer .....                   | 837 |
| Creating a new atdialer dialer .....               | 838 |
| Modems that vary line speed .....                  | 839 |
| Editing the connect speed strings .....            | 839 |
| RTS/CTS flow control .....                         | 840 |
| RTS/CTS on outgoing calls .....                    | 840 |
| RTS/CTS on incoming calls .....                    | 840 |
| atdialer creation procedure .....                  | 841 |

## Chapter 17

# Helpful Hints

843

---

|                                               |     |
|-----------------------------------------------|-----|
| Setting the broadcast address .....           | 843 |
| Problem with WD8003 card .....                | 844 |
| Making remote backups .....                   | 844 |
| Backing up files or filesystems .....         | 845 |
| Restoring a backup .....                      | 846 |
| Differences in sendmail implementations ..... | 846 |
| Setting up user equivalence .....             | 847 |

---

|             |     |
|-------------|-----|
| Index ..... | 997 |
|-------------|-----|

## Chapter 1

# Networking overview

---

This guide provides background and reference material that helps you configure and maintain the various networking services of SCO® Open Desktop®/SCO Open Server™.

This chapter describes networking in general and the SCO Networking Services in particular. After you read this chapter, you will have a better understanding of the components that make up the networking package, and you can pick and choose which components you want to configure. We strongly recommend that you read this chapter before installing any networking software.

The remaining chapters in this guide describe the networking products that make up the SCO system basic networking package: SCO TCP/IP, SCO NFS™, SCO LAN Manager Client, UUCP, and MMDF.

- Chapters 2 through 8 describe TCP/IP, including basic maintenance, configuring serial line communications, using the BIND name server, setting up gateways, using the Simple Network Management Protocol (SNMP), setting up remote printing, and using network time protocols.
- Chapters 9, 10, and 11 cover NFS, including basic maintenance, enabling and using the Network Information Service (NIS), and configuring the Automounter.
- Chapter 12 describes LAN Manager Client and describes how to share resources on your network between machines running LAN Manager or SCO XENIX-NET.
- Chapter 13 covers uucp and its use in transferring files and performing remote command execution.
- Chapters 14 and 15 describe the two supported mailers provided with your SCO system: MMDF and **sendmail**.

- Chapter 16 explains how to configure and administer modems and modem lines on your system.
- Chapter 17 provides information on how to solve some typical networking problems.

Within each of these chapters, you are often referred to manual pages for further information. All of the networking manual pages referred to in this guide are available online through use of the `xman(X)` or `man(C)` commands.

# *Introduction to networking*

---

Networking, simply put, is connecting your computers together so they can share information. Effective networking increases productivity by using computer resources, such as files, printers, and memory, more efficiently. A network puts the power of all of your system's hardware and software at your fingertips.

Although there are many different types of networks, they fall into two general categories: local area networks (LANs) and wide area networks (WAN).

A LAN connects computers that are in the same office or in adjacent buildings. All the computers on a LAN are connected to a single cable. A computer on a LAN can communicate directly to any other computer on that LAN. One LAN may also be connected to another LAN via a gateway computer.

A WAN connects computers that can be as close as several hundred feet to as far as across the globe. These connections are made using phone lines and sometimes satellite connections, if the distance is great enough. Sometimes a computer must go through one or more computers, or gateways, to reach the one with which it wants to communicate.

Most networks are a combination of local and wide area networks. Figure 1.1 displays a portion of a typical local area network. It includes several client computers, a server computer, a printer that is accessible to any machine on the network, an Ethernet™ cable connecting the machines, and a computer running SCO Open Desktop/SCO Open Server, which may be a client, a server, or both.

Servers, often the most powerful computers on the network, store data that they make available to clients, other machines on the network that have access to the servers' resources. You can have one or more servers on a network, and a machine can be both a client and server. For example, one machine can serve personnel information while another serves sales data. Each machine is, therefore, a server, but each may also be a client to the other machine's data.

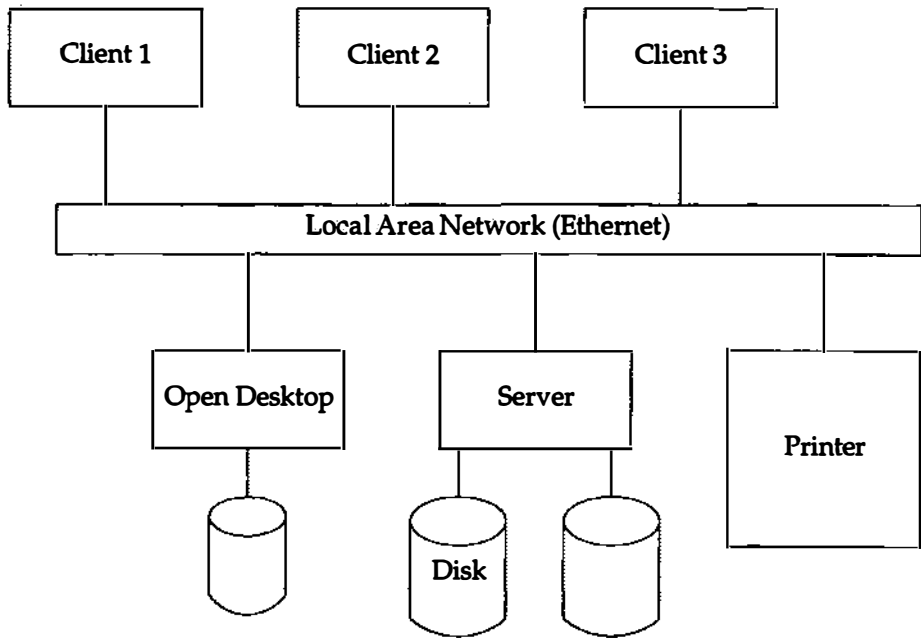


Figure 1-1 Sample Network

## How the network works

A network, in the physical sense, consists of cables or phone lines. These lines connect the computers, and networking cards provide the means to talk across them. However, a network is not useful unless it has programs on each computer that let humans access the various computers on the network.

Computers on a network have agreed ways of communicating called protocols. Protocols dictate which signals computers use across cables, how they tell one another that they have received information, and how they exchange information.

Protocols are more accurately termed protocol suites or protocol families. This subtle shift in terminology reflects the fact that the communications functions are complex and are usually divided into independent layers, also called levels. The protocol associated with each layer communicates with only the layers immediately above and below it, and assumes the support of underlying layers.

In protocol suites, lower layers are closer to the hardware and higher layers are closer to the user. The number of layers and tasks that the layers perform depends on who defines them. The following sections describe several popular definitions of protocols and how they relate to each other.

## OSI

---

The International Organization for Standards (ISO) has a seven-layer protocol called the Reference Model of Open Systems Interconnection (OSI), as shown in Table 1-1. OSI also serves as a basis for the X.25 protocol.

**Table 1-1 ISO Model**

| Layer | Name         |
|-------|--------------|
| 7     | Application  |
| 6     | Presentation |
| 5     | Session      |
| 4     | Transport    |
| 3     | Network      |
| 2     | Data Link    |
| 1     | Physical     |

The layers operate as follows:

| Layer        | Task                                                                                                                                        |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Application  | defines how programs can communicate with each other                                                                                        |
| Presentation | performs any necessary data conversion                                                                                                      |
| Session      | establishes an enhanced connection (session) with other machines                                                                            |
| Transport    | makes sure information exchanged between computers arrives intact and without errors                                                        |
| Network      | makes sure information coming from one computer arrives at the correct destination                                                          |
| Data Link    | makes sure information gets from one end of the cable to the other intact                                                                   |
| Physical     | defines the way information travels on a cable. A common physical-level protocol is the IEEE 802.3 protocol, of which Ethernet is a subset. |

## TCP/IP

---

Transmission Control Protocol with Internet Protocol (TCP/IP) has four software layers built on an underlying hardware layer. Its model is shown in Table 1-2.

**Table 1-2 TCP/IP Model**

| Layer | Name              |
|-------|-------------------|
| 4     | Application       |
| 3     | Transport         |
| 2     | Internet          |
| 1     | Network Interface |

The layers operate as follows:

| Layer             | Task                                                |
|-------------------|-----------------------------------------------------|
| Application       | accesses the Internet, and sends and receives data  |
| Transport         | provides communication between application programs |
| Internet          | takes care of communication among machines          |
| Network Interface | accepts and transmits data over the network         |



## Correspondence of models

Table 1-3 shows how the ISO and TCP/IP models correspond to each other.

**Table 1-3 Protocol Families**

| ISO   |               | TCP/IP            |       |
|-------|---------------|-------------------|-------|
| Layer | Functionality | Functionality     | Layer |
| 7     | Application   | Application       | 4     |
| 6     | Presentation  |                   |       |
| 5     | Session       |                   |       |
| 4     | Transport     | Transport         | 3     |
| 3     | Network       | Internet          | 2     |
| 2     | Data Link     | Network Interface | 1     |
| 1     | Physical      | Hardware          | —     |

## Designing or adding to your network

---

Your SCO system may be a part of an entirely new network, or it may become a part of a network that already exists. In either case, you need to make several decisions about your system:

- With what other computers does it need to communicate?
- Will it serve as a client, a server, or both?
- Who will use this machine, and what sort of access do they need?

## Common networking administration tasks

---

After you decide how your SCO system fits into the network, you need to install and configure the appropriate Networking services as described in the *SCO Open Desktop/SCO Open Server Installation and Upgrade Guide*. You also

need to update the networking files on other machines so that they know of the new machine's existence. This configuration ensures, among other things, that:

- all machines on the network know each other's names and addresses
- individual users will have access to files and accounts on various machines
- electronic mail is routed correctly
- the network runs at peak efficiency

Common tasks that you will perform to ensure these goals include:

- installing and maintaining networking hardware and software
- assigning names and addresses to each computer and device on the network
- assigning names and identification numbers (IDs) to network users and groups
- performing the commands required to share, remove, and restrict resources
- updating all appropriate networking files on your network's machines

The remainder of this chapter describes the various Networking services used by your SCO system, and provides pointers to the rest of this guide.

## Networking services

---

The networking package consists of several services that allow your computer to connect to both local and wide area networks.

- TCP/IP provides various protocols for networking communications that other networking packages, such as NFS, use. The TCP/IP package also provides end-user programs such as `telnet` and `ftp` that enable remote logins and file transfer between systems running TCP/IP.
- NFS allows you to export filesystems to the network so that users on other computers can use them as if they were local, and to import filesystems from remote servers to your SCO system client.
- LAN Manager Client provides a way to access resources (files, programs, printers, and so on) remotely from other machines running compatible versions of LAN Manager and XENIX-NET.
- UUCP enables you to execute commands remotely and to transfer files and mail between remote UNIX systems connected by serial lines.
- MMDf controls electronic mail communications between machines running the UNIX Operating system.

The remainder of this chapter introduces each of these services. It briefly explains the issues involved in configuring and maintaining each service and describes many of the files and utilities that each service uses.

## Introducing TCP/IP

---

TCP/IP is the set of protocols and programs used by your SCO system to interconnect computer networks and to route traffic among different types of computers. "TCP" stands for Transmission Control Protocol, and "IP" stands for Internet Protocol. These protocols describe allowable formats, error handling, message passing, and communication standards. Computer systems that use TCP/IP are thus able to speak a common language, despite any differences in the hardware and software of the various machines.

Many large networks conform to these protocols, including the DARPA Internet (Defense Advanced Research Projects Agency Internet). A variety of universities, government agencies, and computer firms are connected to an internetwork that follows the TCP/IP protocols. Thousands of machines are connected to this internet, or network of networks. Any machine on the internet can communicate with any other. Machines on the internet are referred to as *hosts* or *nodes*, and are defined by their internet (or IP) address. Defining an internet address is described later in this section.

TCP/IP provides the basis for many useful services, including electronic mail, file transfer, and remote login. Electronic mail is designed to transfer short text files. The file transfer application programs transfer very large files containing programs or data. They also provide security checks controlling file transfer. Remote login allows users on one computer to log in at a remote machine and carry on an interactive session. The TCP/IP programs that facilitate these services are described in detail in the SCO Open Desktop/SCO Open Server *User's Guide*.

### *The internet protocol (IP)*

The Internet Protocol, IP, defines a data delivery system wherein the sending and receiving machines are not necessarily directly connected. IP splits data into packets of a given size, which are then forwarded to the receiving machine via the network. These individual packets of data (often called *datagrams*) are routed through different machines on the internet to the destination network and receiving machine. A particular set of data, such as a file, may be broken up into several datagrams that are sent separately. When you use IP to forward datagrams, individual datagrams may or may not arrive, and they probably will not arrive in the order in which they were sent. TCP adds the reliability that IP lacks.

A datagram consists of header information and a data segment. The header information is used to route and process the datagram. Datagrams may be further fragmented into smaller pieces, depending on the physical requirements of the networks they cross. For example, when a gateway sends a datagram to a network that cannot accommodate the datagram as a single

packet, the datagram must be split into pieces that are small enough for transmission. The datagram fragment headers contain the information necessary to reassemble the fragments into the complete datagram. Fragments do not necessarily arrive in order; the software module implementing the IP protocol on the destination machine must reassemble the fragments into the original datagram. If any fragments are lost, the entire datagram is discarded.

### ***The transmission control protocol (TCP)***

The Transmission Control Protocol, TCP, works with IP to provide reliable delivery. It provides a means to ensure that the various datagrams making up a message are reassembled in the correct order at their final destination and that any missing datagrams are resent until they are correctly received.

The primary purpose of TCP is to avoid the loss, damage, duplication, delay, or misordering of packets that can occur under IP. Also, security provisions such as limiting user access to certain machines can be implemented through TCP.

TCP provides reliability using checksums (error detection codes) on the data, sequence numbers in the TCP header, positive acknowledgment of data received, and retransmission of unacknowledged data.

### ***Other TCP/IP protocols***

The protocols listed in Table 1-4 are provided as part of TCP/IP:

**Table 1-4 Additional TCP/IP protocols**

| Protocol                                   | Purpose                                                             |
|--------------------------------------------|---------------------------------------------------------------------|
| Address Resolution Protocol (ARP)          | ARP translates between DARPA Internet and Ethernet addresses.       |
| Internet Control Message Protocol (ICMP)   | ICMP is an error-message and control protocol used by TCP/IP.       |
| Point-to-Point Protocol (PPP)              | PPP provides both synchronous and asynchronous network connections. |
| Reverse Address Resolution Protocol (RARP) | RARP translates between Ethernet and DARPA Internet addresses.      |
| Serial Line Internet Protocol (SLIP)       | SLIP enables IP over serial lines.                                  |
| Simple Mail Transport Protocol (SMTP)      | SMTP is used by MMDF to send mail via TCP/IP.                       |

*(Continued on next page)*

**Table 1-4 Additional TCP/IP protocols**  
(Continued)

| Protocol                                  | Purpose                                                                                                                                                                                               |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Simple Network Management Protocol (SNMP) | SNMP is the protocol used to perform distributed network management functions via TCP/IP.                                                                                                             |
| User Datagram Protocol (UDP)              | UDP provides data transfer without many of the reliable delivery capabilities of TCP. UDP is less CPU-intensive than TCP, and is useful when guaranteed data delivery is not of paramount importance. |

These protocols are described in further detail later in this guide.

### ***TCP/IP end-user commands***

Several TCP/IP commands, described in detail in the SCO system *User's Guide*, provide end users with networking capabilities. Table 1-5 is a partial list of these commands:

**Table 1-5 TCP/IP Commands**

| Command | Purpose                                                                                                    |
|---------|------------------------------------------------------------------------------------------------------------|
| ftp     | file transfer between machines running TCP/IP; these machines may or may not run the same operating system |
| rcmd    | remote command execution on another UNIX machine                                                           |
| rcp     | file copying between two UNIX machines                                                                     |
| rlogin  | remote login on another UNIX machine                                                                       |
| ruptime | status display of local network machines                                                                   |
| rwho    | displays list of users logged on to local network machines.                                                |
| telnet  | remote login on a machine running TCP/IP; these machines may or may not run the same operating system      |

## ***Configuring TCP/IP***

This section provides information on software and hardware prompts you need to answer as you configure TCP/IP. We strongly recommend that you read and understand this section before you fill in the configuration checklists located in the *Installation and Upgrade Guide*. Installation prompts include:

- system's host name and domain name
- Internet address(es) for each driver, adapter, or serial line
- broadcast address

- netmask
- hardware interrupt vectors, base memory addresses, and RAM buffer sizes

## *System name*

Your system name, or host name, should be unique on your network. It can consist of lowercase letters and numbers, must begin with a letter, and should be no longer than eight characters. mail and other programs use the system name to identify the correct data destination. Here are some sample valid machine names: *scosysv*, *tcpdev*, *account1*.

## *Driver type*

The driver is the software that allows your networking cards or hardware to interact with TCP/IP. Each card, adapter, slip line or ppp line that you use must be uniquely associated with a particular device driver. You can install up to four Ethernet cards of one type, up to two Token Ring adapters, and up to eight serial line interfaces (four SLIP and four PPP), but you can only configure one driver at a time. When you are prompted for the driver type, choose the type you want to configure.

## *Interrupt vector*

Each driver on your system, including those for network cards and SLIP lines, must have its own interrupt vector, or IRQ. This vector must not be used by any other device on the system. Refer to your networking hardware documentation to determine what vectors the hardware supports. In addition, the **hwconfig(ADM)** and **vectorsinuse(ADM)** programs list the hardware already installed on your system and what vectors are already in use, respectively.

Your networking hardware might be pre-configured to use a particular vector. If you want to change this vector setting, you might also need to change the physical jumper settings on the board or run a setup program provided with the board.

**NOTE** A number of networking cards are pre-configured to use interrupt vector 3. Your operating system has reserved IRQ3 for the sio (serial input-output) device. You can either disable this device during your **netconfig** session, or choose another vector.

## *I/O base address*

Each hardware driver on your system that performs I/O (input/output) needs a unique memory base address so that the system can locate it. This memory address is a three- or four-digit hexadecimal number, must match the settings on the board, and must not conflict with any other hardware on your system. Valid base addresses are displayed when you configure your card.

### *Thick/thin cable*

Some networking cards use thick, rather than thin, networking cable.

- Thin cable provides a direct connection to the network without the use of a transceiver. Most installations use thin cable.
- Thick cable connects your networking card to a transceiver, which in turn connects to the Ethernet cable.

### *RAM buffer size and base address*

Several networking cards require a designated space in RAM to do buffering; you need to specify this address (as a five-digit hexadecimal number) and, if necessary, configure the buffer size.

**NOTE** The `wdnsetup` command is used to change these values for some Western Digital cards. For more information on this command, see the `wdnsetup(ADM)` manual page.

### *Domain name*

The MMDF mail router uses the domain name to route messages, such as mail, from machine to machine. The domain name allows your network to fit into a hierarchical network structure composed of commercial organizations (.COM), educational institutions (.EDU), the government (.GOV), the military (.MIL) or miscellaneous organizations (.ORG). Sample domain names are *sco.COM* (the domain name used by SCO) and *berkeley.EDU* (the domain name used by the University of California at Berkeley).

Base your domain name choice on the following:

- If other machines on your network already use a domain name, use the same name for the machine you are installing.

- If you are creating a new domain and want to use BIND to connect to the outside world, you need to register the name with the appropriate network (DARPA Internet, CSNET, or BITNET). To register a domain name, write to:

Network Solutions  
 Attn: InterNIC Registration Services  
 505 Huntmar Park Drive  
 Herndon, VA 22070

- If you are creating a new domain and are not currently connecting to an outside network, use the name .UUCP.
- If you will never attach to a network outside your company, choose *company.COM*.

## IP address

The IP address identifies and differentiates your machine from all others on the network. It consists of a 32-bit binary number that is usually displayed as four octets expressed in decimal and separated by periods. You must have a unique IP address for each machine on your network. In addition, if your machine serves as a *router* to another network, it contains two or more network cards and belongs to two or more networks. In this case, you must assign each card a unique IP address on the appropriate network.

**NOTE** The IP address differs from an Ethernet address in that it is configurable. An Ethernet address is a 6-byte address that is unique to each physical Ethernet card. This non-configurable address is assigned by the card manufacturer.

The IP address consists of two parts: a network address that identifies the network and a host address that identifies the particular host, or node. Table 1-6 shows an IP address in binary form, as binary octets, as decimal octets, and as it appears in standard notation.

**Table 1-6 IP address derivation**

|                 |                                                                 |                 |                 |                 |
|-----------------|-----------------------------------------------------------------|-----------------|-----------------|-----------------|
| binary (32-bit) | 1 0 0 0 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 |                 |                 |                 |
| binary (octets) | 1 0 0 0 0 1 0 0                                                 | 1 0 0 0 1 1 1 1 | 0 0 0 0 0 0 1 0 | 0 0 0 0 0 0 1 0 |
| decimal octets  | 132                                                             | 147             | 2               | 2               |

IP address (in standard notation) = 132.147.2.2

Several classes of TCP/IP networks are available, each based on the number of hosts a network needs. Network classes supported by SCO are Class A, B, and C. Use the smallest network class that can accommodate all of your network's hosts. Most TCP/IP installations use Class C, but some larger installations might need to use Class B.



### ***Source IP address***

Enter the IP address for this host (this end of the serial line). For more information on determining IP addresses, see “IP address” earlier in this chapter.

### ***Destination IP address***

Enter the IP address for the remote host (the opposite end of the line).

### ***Baud rate***

Enter the baud rate at which data is transmitted. The default is 9600.

### ***SLIP netmask***

Choose a netmask for this SLIP line. For more information on netmasks, see the section “netmask setting” earlier in this chapter.

## ***Maintaining TCP/IP***

---

After you install TCP/IP, you may never need to alter the TCP/IP configuration again. However, there are some common tasks that occur if you want to customize or add to your network. These are described briefly here and in detail later in this guide.

### ***Adding hosts***

The */etc/hosts* file is a list of hosts on the network. Network library routines and server programs use this file to translate between host names and Internet addresses when the BIND (Berkeley Internet Name Domain) name server is not being used.

To add a machine to the network, you must add an entry to all of the */etc/hosts* files on the local network. Refer to the *hosts(SFF)* manual page for a description of the file format.

### ***Configuring the name domain server***

The Berkeley Internet Name Domain Server (BIND) provides a distributed lookup system for host names and addresses. Enabling BIND overrides the default network information file, */etc/hosts*. For more information, see the chapter “Configuring the BIND name server” later in this guide.

### ***Setting up routing tables***

Routing tables provide the information needed to route packets to their destinations properly. For descriptions of several possible approaches to maintaining routing information, see the chapter “Gateways and routing” later in this guide. In addition, the chapter “TCP/IP network administration” contains a section on obtaining information about the system routing tables.

## *Establishing user equivalence*

You can control who has access to a machine through the network by establishing user equivalence within the */etc/hosts.equiv* and *.rhosts* files. The **rlogin**, **rcp**, and **rcmd** commands use these files to verify access privileges. For information on how to use these files, see the section “Network databases” in the chapter “TCP/IP network administration” later in this guide. You can also refer to the *hosts.equiv*(SFF) manual page for a description of the file format. A note in the “Helpful Hints” chapter also discusses user equivalence.

## *Setting up anonymous ftp*

You can set up a public **ftp** account on your system that allows remote users to transfer files anonymously from restricted, public directories on your system. For information on the */etc/ftpusers* file, and a description of how to set up the public **ftp** account, refer to the section “Network databases” in the chapter “TCP/IP network administration” later in this guide.

## *Altering installation parameters*

You can change many of the settings that you set during TCP/IP installation by altering the appropriate system files (such as */etc/hosts* and device driver files) with a text editor or with an appropriate utility, such as **netconfig**, **route**, or **mkdev**. The use of such files and utilities, which is documented in the chapter “TCP/IP network administration,” is always preferable to reinstalling the software.

## *Tuning kernel parameters*

You may need to tune kernel parameters by increasing or decreasing STREAMS buffers and other parameters used by TCP/IP. Several utilities, including **netstat**, **configure**, and **netconfig**, help you fine-tune your system to enhance networking performance. These utilities are described in the chapter “TCP/IP network administration.”

## *Monitoring TCP/IP status*

You can use the **netstat** command to display Internet connections, current Internet activity, routing tables, and error messages, among other useful information. In addition, you can use the Simple Network Management Protocol (SNMP) commands and utilities to further monitor and troubleshoot your network. For more information, see the chapter “Configuring and using SNMP” later in this guide.

## *Enabling remote printing*

You can enable the remote printing daemon, **lpd**, to allow print jobs to be sent over the network to remote printers, or to make a printer attached to your computer available to the network. For information on **lpd** and its associated files, see the chapter “Remote line printing” later in this guide.

## Introducing NFS

---

SCO Network File System (NFS) is a product that enables you to export and mount filesystems across a network. This allows users on a local machine, or client, to access specified files and directories from a remote machine, or server, without the time-consuming process of remote logins or machine-to-machine file copying.

NFS also contains the Network Information Service (NIS), which enables you to perform common account administration tasks from one NIS server.

Because NFS fits into the application layer of TCP/IP, it can communicate between many different types of hardware and operating systems, as long as these systems run TCP/IP.

NFS makes use of the following files, commands, and daemons, each of which are described in detail in Chapter 9, "Administering NFS," of this guide.

- The file */etc/exports* lists the exportable filesystems on a given server.
- The file */etc/hosts* informs TCP/IP, and hence NFS, how to find a given server or client.
- The file */etc/default/filesys* lists the remote filesystems a client may mount and on what server the client may find them.
- The **mount** command mounts an exported filesystem on the client. This command invokes the **mountd** and  **nfsd** daemons.

## Maintaining NFS

---

If you want to export filesystems with NFS, you must create */etc/exports* after installing your SCO system. A sample file is shown in Chapter 9, "Administering NFS." In addition to creating */etc/exports*, you also must ensure that the following conditions are true:

- The */etc/hosts* file on each server and client must contain the names and IP addresses of each server and client you want to access.
- The */etc/default/filesys* file on each client must contain the names and locations of all of the filesystems that you want to access.

After you edit these files to satisfy these conditions, you may want to fine-tune NFS as shown in Chapter 9.

## Introducing LAN Manager client

---

LAN Manager Client is a SCO system program that allows other computers on a LAN that run compatible servers to share resources. Resources can be any of the following:

- files
- directories
- programs
- printers
- plotters
- tape drives

LAN Manager is designed so that users can access files and printers on other computers running LAN Manager simply by typing a command on their own computer. Because LAN Manager runs on DOS and UNIX systems as well as your SCO Open Desktop/SCO Open Server, you can have PCs, multiuser UNIX systems, and SCO system workstations sharing resources on the same LAN.

To use LAN Manager correctly between your SCO Open Desktop/SCO Open Server and UNIX systems, you must maintain user (UID) and group (GID) consistency between the machines on your network. If you are connecting existing computers in a network for the first time, be certain that user IDs are consistent from computer to computer. If you are adding your SCO system to an existing network, you should create user accounts and groups with IDs that match those found throughout the network. User IDs are defined by the **sysadmsh(ADM) Accounts ⇨ User ⇨ Create** selection and placed in UNIX system database files, including */etc/passwd*. For information on adding user accounts, see the "Administering User Accounts" chapter in the *Administering Operating System Services* portion of this guide. For information on maintaining ID consistency, see the **passwd(C)**, and **passwd(F)** manual pages. When you create new user accounts or change account information, keep in mind the following:

- Always use the **sysadmsh(ADM) Accounts ⇨ User ⇨ Create** selection to administer accounts. Never edit */etc/passwd* with a text editor. This could damage the password database information and cause your system to refuse further logins.
- User ID numbers must not be duplicated among different users. An individual user can have several different accounts on different machines, but to maintain user equivalency each account must have the same user name and ID number.

## Configuring LAN Manager Client

---

When you install LAN Manager Client, the **custom** utility configures LAN Manager Client and adds your computer to the network. Before LAN Manager Client is ready to use, you must configure it with **netconfig**(ADM) as described in the *Installation and Upgrade Guide*.

When you enable LAN Manager Client with **netconfig**, you are prompted to confirm the hostname and confirm that LAN Manager Client starts up each time the system is brought into multiuser mode. The most important prompt, however, asks you to provide a NetBIOS scope identifier for your LAN Manager network.

All machines on your network must use the same scope identifier to successfully communicate with one another. Choose one based on the following guidelines:

- If you are adding this machine to an existing network, use the same scope identifier that is used on any other machine on the network. On a SCO UNIX or SCO Open Desktop/SCO Open Server machine, the scope identifier is found in the file */etc/default/nbconf* as the variable **NB\_SCOPE**.
- If you are creating a new network that contains only SCO UNIX or SCO Open Desktop/SCO Open Server machines, use the domain name (such as *sco.COM*) minus its suffix. For example, if your domain name is *mycomp.COM*, enter **mycomp**.
- If you are creating a new network that also contains DOS or OS/2 machines, it is a good idea to set the scope identifier to **NULL**.

**NOTE** To enter a NULL scope identifier during **netconfig** configuration, enter two double quotes ("") at the prompt.

You can enter a scope identifier or choose the default during your **netconfig** session, or you can change the scope identifier later by editing the file */etc/default/nbconf*. To change the scope identifier in this file, change the **NB\_SCOPE** variable to the desired scope. To set this variable to **NULL**, edit the variable line so that it reads:

```
NB_SCOPE=""
```

If you use this procedure to change **NB\_SCOPE**, you must reboot the machine before this change takes effect.

## Maintaining LAN Manager Client

---

A common activity that you may perform as a system administrator is configuring LAN Manager Client so that it runs more efficiently. You do so in two ways: by altering system-wide configuration parameters and relinking the

kernel, or by changing default values found in the networking startup file, */usr/lib/xnet/xnetrc*. For example, you might decrease the number of network buffers available, making more memory available for user programs. For information on these maintenance issues, refer to Chapter 12, "LAN Manager client administration."

## Introducing UUCP

---

UUCP, the UNIX-to-UNIX copy program, is used when your SCO system's network connection consists of a serial line and modem connection to another machine. If your network consists of an Ethernet LAN, with or without an Internet connection, you do not need to use UUCP, as it duplicates many features found in LAN Manager Client and TCP/IP.

The following UUCP features are available:

- file transfer between your machine and a remote UNIX machine, or between remote UNIX machines, with the `uucp` command; this command is comparable to the `ftp` and `rcp` commands
- remote command execution with the `uux` command; this command is comparable to the `rcmd` command
- mail delivery to and from remote sites with the `mail` command via the UUCP modem and phone line connection

UUCP spools requests for communication until contact is made with a remote system. You can configure your SCO system to communicate with other machines rarely, daily, hourly, or even on demand.

UUCP consists of a variety of administrative and end-user programs that help you maintain and use UUCP. These programs are described in detail in Chapter 13, "Building a Remote Network with UUCP," in this guide.

## Configuring UUCP

---

To set up UUCP for use, do the following:

1. Attach a modem and phone line to a serial port on your computer.
2. Use `uucinstall` to configure UUCP on your system by choosing a site name, updating host lists, and configuring dial-in/dial-out lines.

After you perform these actions, you can use UUCP. Several `uudemon` shell scripts and the `uulog` program help you maintain UUCP after configuration by polling sites automatically, displaying status information, cleaning UUCP directories, and so forth. For more information on these programs, refer to Chapter 13.

## Introducing MMDF

---

Your SCO system MMDF, the Multi-channel Memorandum Distribution Facility, to route mail locally and to and through channels (such as UUCP) that offer MMDF support. Here are the key facets of MMDF:

- Domain names are used to route messages from machine to machine.
- Channels, such as UUCP, SMTP, or Micnet, are the programs that MMDF uses to move mail from one location to another. Aliases allow you to provide your users with shorthand ways of specifying addresses. For example, the address *info-p1-outbound@berkeley.EDU* could be aliased to *info*.
- A configuration file, */usr/mmdf/mmdftailor*, pulls all of the MMDF pieces together by specifying the local machine's name, domain, company, and UUCP name, as well as describing how, when, and where to contact other machines.

## Configuring MMDF

---

After the appropriate files are installed on your system, you need to run `mkdev mmdf` to configure MMDF. Refer to Chapter 14, "Setting up electronic mail," for more information.

## Other networking software

---

Your SCO system provides two other types of networking software that do not fall directly under any of the packages described earlier in this chapter. They are **sendmail** and **cu**.

**sendmail** is another mail routing and distribution program. We recommend that you use MMDF, as it provides an enhanced feature set and is easier to configure than **sendmail**. If you do decide to use **sendmail**, complete instructions on installation and configuration are found in Chapter 15, "Sendmail administration," of this guide.

**cu(C)** allows you to dial in and dial out of your SCO Open Desktop/SCO Open Server computer by means of a standard telephone line and modem. In doing so, you can emulate a terminal session on a remote machine, or dial into your SCO Open Desktop/SCO Open Server machine from a remote terminal or computer. Chapter 16, "Using modems," explains how to install and configure the appropriate hardware on your system to perform these actions.

## Chapter 2

# TCP/IP network administration

---

This chapter covers topics related to setting up and administering your TCP/IP network. When you installed and configured your system, many of these tasks were performed automatically by the **netconfig**(ADM) command to configure a basic networked system. For more information on **netconfig**, see the **netconfig**(ADM) manual page. If you want to customize your installation or troubleshoot your network, you should read this chapter.

Topics covered include modifying tunable kernel parameters, establishing user equivalence, setting up anonymous ftp, and administering pseudo ttys. In addition, if your network is not performing well, the section “Network tuning and troubleshooting” at the end of this chapter provides helpful suggestions.

## Kernel configuration

---

Table 2-1 lists the tunable kernel parameters for this release of TCP/IP. These parameters are configured by default to work efficiently in most situations. However, if your system has an unusual configuration, you may need to tune these parameters. Please read the descriptions that follow Table 2-1 before you change any tunable kernel parameters. The following steps explain how to change the parameters:

1. Determine the name of the file to edit by looking in the second column of Table 2-1. The complete filename will be `/etc/conf/pack.d/` plus the value from the table. For example, to change the value of **nb\_sendkeepalives**, edit the file `/etc/conf/pack.d/nb/space.c`. Note that the directories in *pack.d* only exist if they have been configured. For example, *ppp* will only appear if PPP has been configured using **netconfig**.



2. Make a backup copy of the file in case you need to restore the original values later.
3. Using a text editor, such as `vi`, locate the line containing the variable and change the supplied value. Where the current value is a constant that is defined in a header file, do not edit the header file. Instead, replace the constant with the new value. Where the parameter is boolean (that is, either 0 or 1), change the current value to the opposite boolean value (for example, change 0 to 1).
4. Relink the kernel using the `sysadmsh` selection, `System ⇨ Configure ⇨ Kernel ⇨ Rebuild ⇨`. Follow the prompts; respond `y` both when asked if you want to have the new kernel boot by default and when asked if you want to rebuild the kernel environment.
5. Boot the new kernel using the `sysadmsh` `System ⇨ Terminate` selection to shut the system down. Press `(Return)` when the reboot message is displayed.

**NOTE** Randomly changing these parameters can seriously harm your system. Make absolutely sure that you understand the implications of any changes you make.

**Table 2-1 Tunable kernel parameters**

| Parameter                      | File                      | Description                        | Default |
|--------------------------------|---------------------------|------------------------------------|---------|
| <code>ahdlcmtn</code>          | <code>asyh/space.c</code> | asyh MTU                           | 296     |
| <code>icmp_answermask</code>   | <code>ip/space.c</code>   | answer subnet mask requests        | 0       |
| <code>ipforwarding</code>      | <code>ip/space.c</code>   | forward datagrams                  | 0       |
| <code>ipprovcnt</code>         | <code>ip/space.c</code>   | # of IP interfaces                 | 16      |
| <code>ipsendredirects</code>   | <code>ip/space.c</code>   | send ICMP redirects                | 0       |
| <code>NB_NNAMETAB</code>       | <code>nb/space.c</code>   | # of NetBIOS names                 | 64      |
| <code>NB_NREQ</code>           | <code>nb/space.c</code>   | # of NetBIOS Control Blocks NCBs   | 64      |
| <code>NB_NSESSION</code>       | <code>nb/space.c</code>   | # of NetBIOS sessions              | 64      |
| <code>nb_sendkeepalives</code> | <code>nb/space.c</code>   | NetBIOS level keepalives on/off    | 0       |
| <code>pppmtu</code>            | <code>ppp/space.c</code>  | PPP MTU                            | 296     |
| <code>slmtu</code>             | <code>slip/space.c</code> | SLIP MTU                           | 296     |
| <code>subnetsarelocal</code>   | <code>ip/space.c</code>   | treat subnets as local             | 1       |
| <code>tcpprintfs</code>        | <code>tcp/space.c</code>  | TCP checksum errorsnotice          | 1       |
| <code>tcp_recvspace</code>     | <code>tcp/space.c</code>  | TCP default receive window size    | 4096    |
| <code>tcp_round_mss</code>     | <code>tcp/space.c</code>  | round MSS down to multiple of 1024 | 1       |
| <code>tcp_ttl</code>           | <code>tcp/space.c</code>  | TCP default TTL                    | 60      |
| <code>udp_ttl</code>           | <code>udp/space.c</code>  | TCP default TTL                    | 30      |

The following list describes how the variables affect TCP/IP and the cases in which each variable needs to be changed.

#### **ahdlcmtu, pppmtu, slmtu**

Change these variables if a machine at the other end of a serial link has a larger Maximum Transmission Unit (MTU) size which cannot be changed. The MTU is the largest amount of data that can be transferred across a given physical network. For LANs, the MTU is determined by the network hardware. For WANs, the MTU is determined by the software.

**pppmtu** controls the Internet Engineering Task Force Point to Point Protocol. **slmtu** controls the SLIP (Serial Line IP) protocol.

#### **icmp\_answermask**

Set to 1 (true) to cause the system to respond to ICMP subnet mask requests. This variable is needed to support diskless workstations.

#### **ipforwarding, ipsendredirects**

Set these to 1 if this machine is to be used as a gateway. **ipforwarding** controls whether the system will forward packets sent to it which are destined for another system. When a packet is forwarded to another system which could have been reached directly by the originating system, an ICMP redirect message will be sent to the originating system if **ipsendredirects** is set to 1 (true).

The **netconfig**(ADM) utility also configures these values when drivers (after the first one) are added. **netconfig** edits *mtune* in the */etc/conf/cf.d* directory. This **netconfig** feature usually makes it unnecessary to change **ipforwarding** and **ipsendredirects** in the *ip/space.c* file.

#### **ipprovcnt**

Set to the number of IP interfaces you have, if you have more than network adapter cards used by IP.

#### **NB\_NSESSION, NB\_NNAMETAB, NB\_NREQ**

These variables are respectively, the number of NetBIOS sessions, NetBIOS names and NetBIOS Control Blocks (NCBs) corresponding to the LAN Manager parameters **MAXSESSIONS**, **NNCB\_NAMES** and **NNCB**. If you increase any one of those LAN Manager parameters above the TCP/IP default, you must change the corresponding TCP/IP parameter accordingly. For example, if you want to increase the LAN Manager parameter **MAXSESSIONS** to 128, you must also change **NB\_NSESSION** in *nb/space.c* to 128 and relink the kernel. Note that if you change the defaults for **NB\_NSESSION** in or **NB\_NREQ** in *nb/space.c*, you should also change **NB\_DFLTSSN** and **NB\_DFLNCB** in */etc/default/nbconf* to correspond with the new TCP/IP NetBIOS driver values.

### **nb\_sendkeepalives**

Set to 1 to turn on NetBIOS level keepalives. When turned on, NetBIOS keepalives are sent periodically on dormant NetBIOS connections. NetBIOS keepalives are independent of TCP/IP keepalives, and are useful for systems that do not use TCP/IP keepalives. This parameter is turned off by default.

### **subnetsarelocal**

Set to 1 to request the network adapter maximum packet size on TCP/IP connections to local subnets. This is useful if the local network is made up of links which have maximum packet sizes greater than or equal to the local adapter. "ICMP Host Unreachable" is generated for local subnet routing failures. When this value is set to 0, the packet size is set to 576 bytes, as specified by RFC 1122.

### **tcpprintfs**

If this parameter is set to 1, TCP/IP warning messages are printed to the console. If the parameter is set to 0, the messages are not printed.

### **tcp\_recvspace**

This parameter affects the size of the TCP receive window. If you need a faster TCP data-receive rate, increase **tcp\_recvspace**. For example, increase this parameter if your machine receives a lot of data from remote sites.

The parameter is set to **TCPWINDOW**. Change **TCPWINDOW**'s value by changing the following line in */etc/conf/pack.d/tcp/space.c*:

```
#define TCPWINDOW 4*1024
```

We recommend that you initially change the value to **24\*1024**. You cannot increase the value beyond **64\*1024**.

Increasing the receive window size can sometimes result in exhausting other resources. After you increase the **TCPWINDOW** value, monitor your **STREAMS** buffers, network card statistics, and protocol-specific statistics. See "Network tuning and troubleshooting" (at the end of this chapter) and the **netstat(TC)** manual page for information on monitoring these resources.

### **tcp\_round\_mss**

If this parameter is set to 1, the TCP/IP Maximum Session Size is rounded down to the nearest multiple of 1024. Setting this parameter to 0 prevents the round-down, letting you more effectively use the data portion of a TCP segment. This potentially decreases the number of segments you need for transferring a given amount of data.

### **tcp\_ttl, udp\_ttl**

Increase these parameters if you have an Internet connection with many hops.

## Setting interface parameters

---

All network interface drivers, including the loopback interface, require that their host addresses be defined at boot time. This is done with `ifconfig(ADMN)` commands included in the `/etc/tcp` shell script. These commands are initially set up based on information you supply during configuration with `netconfig(ADM)`.

`ifconfig(ADMN)` also sets the netmask, broadcast address, and IP address at boot time. If you want to change any of this information, you can edit `/etc/tcp` (after making a backup copy of the file), then reboot your system. Instances in which you may want to do so include:

- changing the netmask and broadcast address to make provisions for a sub-network
- changing the broadcast address scheme from all ones to all zeroes, or vice versa
- changing the IP address of the interface

## Creating a subnetwork

---

In TCP/IP, the DARPA Internet support includes the concept of the subnetwork. This is a mechanism that enables several local networks to appear as a single Internet network to off-site hosts. Subnetworks are useful because they allow a site to hide the local topology, requiring only a single route in external gateways. This also means that local network numbers may be locally administered.

To set up local subnetworks, you first need to know how much of the available address space is to be partitioned. The term "address" is used here to mean the Internet host part of the 32-bit address. Sites with a class A network number have a 24-bit address space with which to work, sites with a class B network number have a 16-bit address space; and sites with a class C network number have an 8-bit address space. To define local subnets you must steal some bits from the local host address space for use in extending the network portion of the Internet address.

This reinterpretation of Internet addresses is done only for local networks. It is not visible to off-site hosts. For example, if your site has a class B network number, hosts on this network have an Internet address that contains the network number, 16 bits, and the host number, another 16 bits. To define 254 local subnets, each possessing at most 255 hosts, 8 bits may be taken from the local part to be used for the subnetwork ID. (The use of subnets 0 and all-1's, 255 in this example, is discouraged to avoid confusion about broadcast

addresses.) New network numbers are then constructed by concatenating the original 16-bit network number with the extra 8 bits containing the local sub-network number.

### Network servers

---

In the UNIX system, most of the server programs are started by a super server, called the “Internet daemon”. The Internet daemon, */etc/inetd*, acts as a master server for programs specified in its configuration file, */etc/inetd.conf*, listening for service requests for these servers, and starting up the appropriate program whenever a request is received. The configuration file includes lines containing:

- a service name (as found in */etc/services*),
- the type of socket the server expects (for example, stream or dgram),
- the protocol used with the socket (as found in */etc/protocols*),
- whether to wait for each server to complete before starting up another,
- the user name under which the server should run,
- the server program’s name, and
- up to five arguments to pass to the server program

Some trivial services are implemented internally in *inetd*(SFF), and their servers are listed as internal. For example, an entry for the file-transfer protocol server would appear as:

```
ftp stream tcp nowait root /etc/ftpd ftpd
```

Consult *inetd*(SFF) for more details on the format of the configuration file and the operation of the Internet daemon.

## Network databases

Several data files are used by the network library routines and server programs. Most of these files are host independent and updated only rarely. Table 2-2 lists the data files used.

**Table 2-2 Network database files**

| File                    | Manual reference | Use                              |
|-------------------------|------------------|----------------------------------|
| <i>/etc/hosts</i>       | hosts (SFF)      | host names                       |
| <i>/etc/networks</i>    | networks (SFF)   | network names                    |
| <i>/etc/services</i>    | services (SFF)   | list of known services           |
| <i>/etc/protocols</i>   | protocols (SFF)  | protocol names                   |
| <i>/etc/hosts.equiv</i> | rshd (ADMN)      | list of "trusted" hosts          |
| <i>/etc/ftpusers</i>    | ftpd (ADMN)      | list of "unwelcome" ftp users    |
| <i>/etc/inetd.conf</i>  | inetd (ADMN)     | list of servers started by inetd |

The files distributed are set up for ARPANET or other Internet hosts. Local networks and hosts should be added to describe the local configuration. Network numbers must be chosen for each Ethernet. For sites not connected to the Internet, these can be chosen more or less arbitrarily; otherwise, the normal channels should be used for allocation of network numbers.

### Establishing user equivalence

User equivalence allows a user on one machine to use **rlogin** to log into an equivalent account on another machine without entering a password. You need to modify several files to establish user equivalence: */etc/hosts.equiv*, *.rhosts*, and */etc/passwd*.

To do so, ensure that entries for the affected user exist on both machines in the following files:

- *.rhosts* and */etc/passwd*
- or
- */etc/hosts.equiv* and */etc/passwd*

In both cases, */etc/passwd* must contain an entry for the user name from the remote machine. If you add an entry to *.rhosts* for a particular account, then user equivalence is established for that account only. If you add an entry in */etc/hosts.equiv* for a host name and an account on that host, then that account has user equivalence for any account (except *root*). If the entry in */etc/hosts.equiv* has only the remote host name, then any user on that host has user equivalence for all local accounts (except *root*). Such a host is considered a "trusted host."

**NOTE** Entries in */etc/hosts.equiv* can create large holes in system security. Be sparing in their use. In most circumstances, it is unwise to create entries that allow all users on remote machines to access all accounts on your local machine.

For example, suppose you have an account under the user name *test1* on machine *admin*. You want to establish user equivalence on the remote machine *systemb*. The administrator for the machine *systemb* must create an account for user name *test1*. They must also include the following entry in the file */etc/hosts.equiv* on *systemb*:

```
admin test1
```

This gives user equivalence for all accounts except *root* to user *test1* on the machine *systemb*. Suppose that *test1* really only needed access to the account *testb* on *systemb*. Then it would be better to remove the above entry from */etc/hosts.equiv* on *systemb* and use the following entry in the file *.rhosts* in the home directory for *testb*:

```
admin test1
```

Note that entries for *.rhosts* must include both the system name and the account name. The file */etc/hosts.equiv* does allow entries for the system name only, as discussed earlier.

If there are entries in both *.rhosts* and */etc/hosts.equiv* for the same machine or machine/account combination, then the entry from */etc/hosts.equiv* determines the extent of user equivalence.

### Setting up anonymous ftp

---

The *ftp* server included in the system provides support for an anonymous *ftp* account. Because of the inherent security problems with such a facility, you should read this section carefully if you want to provide such a service.

An anonymous account is enabled by creating a user called *ftp*. When a client uses the anonymous account, a *chroot*(ADM) system call is performed by the server to restrict the client from moving outside that part of the filesystem where the *ftp* home directory is located. Because a *chroot* call is used, certain programs and files used by the server process must be placed in the *ftp* home directory. Further, you must be sure that all directories and executable images are unwritable.

The following directory setup is recommended:

---

```
csh
cd ~ftp
chmod 555 .; chown ftp .; chgrp ftp .
mkdir bin etc pub shlib dev
chown root bin etc shlib dev
chmod 555 bin etc shlib dev
chown ftp pub
chmod 777 pub
cd bin
cp /bin/sh /bin/ls /bin/pwd .
chmod 111 sh ls pwd
cd ../etc
cp /etc/passwd /etc/group .
chmod 444 passwd group
cd ../shlib
cp /shlib/libc_s .
cd ..
find /dev/socksys -print | cpio -dumpv .
```

---

When local users want to place files in the anonymous area, they must place them in a subdirectory. In the setup here, the directory `~ftp/pub` is used.

Another issue to consider is the `/etc/passwd` file placed here. It can be copied by users who use the anonymous account. They can then try to break the passwords of users on your machine for further access. A good choice of users to include in this copy might be *root*, *daemon*, *uucp*, and the *ftp* user.

Aside from the problems of directory modes and such, the *ftp* server provides a loophole for interlopers if certain user accounts are allowed. The file `/etc/ftpusers` is checked on each connection. If the requested user name is located in the file, the request for service is denied. It is suggested that this file contain at least the following names:

```
uucp
root
```

Accounts with nonstandard shells should be listed in this file. Accounts without passwords need not be listed in this file; the *ftp* server does not service these users.



## Administering pseudo ttys

---

When you install TCP/IP, the `netconfig` command allows you to modify the number of pseudo ttys configured on your system. These pseudo ttys allow outside machines to use `telnet` or `rlogin` to access TCP/IP on your machine. If you want to remove these pseudo ttys, or add more to your system, after installing and configuring TCP/IP, use the `sysadmsh(ADM)` command:

1. Log in to the system as *root* and bring the system into system maintenance mode.
2. Enter the following command at your prompt and press `<Return>`:  
`sysadmsh`
3. Select the following: System ⇌ Hardware
4. Select Pseudo ttys from the point-and-pick list.
5. Enter 1 to add pseudo ttys, or 2 to remove pseudo ttys. If you are adding pseudo ttys, continue with the next step. If you are removing pseudo ttys, skip to step 7.
6. Enter a number of pseudo ttys to create and press `<Return>`. If the number is greater than the current system limit, you are asked if you want to increase the limit. Enter `y` to increase the limit or `n` to cancel the increase.  
 Skip to step 8.
7. You return to the main menu. Enter `q` to quit.
8. Enter a number of pseudo ttys to remove and press `<Return>`.
9. You are prompted to relink the kernel. Do so by entering `y` at the prompt.

**NOTE** If you do not relink the kernel now, the changes you made do not take place, and you return to `sysadmsh`. You must then relink at a later time to effect the changes.

10. The following message appears:

```
The UNIX Operating System will now be rebuilt.
This will take a few minutes. Please wait.
Root for this system build is /.
Do you want this kernel to boot by default? (y/n)
```

Answer `y`.

11. The following prompt appears:

```
Do you want the kernel environment rebuilt? (y/n)
```

Answer **y**. A message appears confirming that the kernel is linked and installed.

12. Exit **sysadmsh** and reboot your system by entering the following command:

```
init 6
```

## Network tuning and troubleshooting

---

It is likely that from time to time you will encounter problems using your network. The first thing to do is check your network connections. On networks such as the Ethernet a loose cable tap or poorly placed power cable can result in severely deteriorated service. The **ping(ADMN)** command is particularly useful for confirming the existence of network connections. If there is no hardware problem, check next for routing problems and addressing problems.

The **netstat(TC)** program can also be helpful in tracking down hardware malfunctions. In particular, look at the **-i**, **-m**, and **-s** options in the manual page. The **netstat(TC)** program also shows detailed information about network behavior. Examples of **netstat** displays appear later in this chapter.

If you think a communication protocol problem exists, consult the protocol specifications and attempt to isolate the problem in a packet trace. The **SO\_DEBUG** option can be supplied before establishing a connection on a socket, in which case the system traces all traffic and internal actions (such as timers expiring) in a circular trace buffer. This buffer can then be printed out with the **trpt(ADMN)** program. Most of the servers distributed with the system accept a **-d** option forcing all sockets to be created with debugging turned on. Consult the appropriate manual pages for more information.

### STREAMS tuning

---

If your system does not have enough STREAMS buffers it will lose connections for no reason, hang on processes that communicate over the network, cause programs that communicate over the network to malfunction, and hang on any process that deals with terminal I/O.

Run **netstat -m** at the prompt to display the current STREAMS buffer use. The FAIL column should have a 0 in every row. If failures are listed in this column, increase the corresponding STREAMS buffers. For example, if you have 40 failures in the NBLK512 row, increase the kernel parameter NBLK512. Start by increasing the number of buffers by 50%. If you still have failures, increase NBLK512 again by 50% of the original value until there are no failures.

To increase STREAMS buffers, use **sysadmsh** and select the following:

System ⇨ Configure ⇨ Kernel ⇨ Parameters

Choose the Streams Data option. You are prompted to enter a value for all STREAMS parameters. Press <Return> for values that you do not want to change. Enter the new value for the STREAMS buffer that you want to increase. You can also use the UNIX Link Kit **configure(ADM)** command to increase STREAMS buffer resources.

## Active connections display

The active connections display is the default display of the `netstat(TC)` command. It displays a line of information for each active connection on the local machine under the following heading.

### ***netstat -a***

Active Internet connections (including servers) are as follows:

---

```
scobox$ netstat -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address Foreign Address (state)
ip 0 0 *. * *. *
tcp 0 0 scobox.telnet scoter.2460 ESTABLISHED
tcp 0 0 *.smtp *. * LISTEN
tcp 0 0 *.1024 *. * LISTEN
tcp 0 0 *.sunrpc *. * LISTEN
tcp 0 0 *.chargen *. * LISTEN
tcp 0 0 *.daytime *. * LISTEN
tcp 0 0 *.time *. * LISTEN
tcp 0 0 *.domain *. * LISTEN
tcp 0 0 *.finger *. * LISTEN
tcp 0 0 *.exec *. * LISTEN
tcp 0 0 *.ftp *. * LISTEN
tcp 0 0 *.telnet *. * LISTEN
tcp 0 0 *.login *. * LISTEN
tcp 0 0 *.shell *. * LISTEN
tcp 0 0 scobox.listen *. * LISTEN
tcp 0 0 scobox.nterm *. * LISTEN
tcp 0 0 *. * *. * CLOSED
udp 0 0 *.1035 *. *
udp 0 0 *.1034 *. *
udp 0 0 *.1033 *. *
udp 0 0 *.1032 *. *
udp 0 0 *.2049 *. *
udp 0 0 *.1028 *. *
udp 0 0 *.sunrpc *. *
udp 0 0 scobox.domain *. *
udp 0 0 localhost.domain *. *
```

---

### *Descriptions of the display headings*

|                 |                                                                                                                                                                                |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Proto           | protocol used in the connection                                                                                                                                                |
| Recv-Q          | receive queue. This is the number of received characters (bytes) of data waiting to be processed.                                                                              |
| Send-Q          | send queue. This is the number of characters (bytes) of data waiting to be transmitted.                                                                                        |
| Local Address   | port number of the local connection, displayed symbolically. The port numbers are taken from the <i>/etc/services</i> file.                                                    |
| Foreign Address | port number of the remote connection, displayed symbolically. The port numbers are taken from the <i>/etc/services</i> file.                                                   |
| State           | current state of the connection. Each protocol has its own set of states. For the protocol-dependent states that can be displayed, see the appropriate protocol specification. |

## *Interfaces*

---

The interface display describes activities on all the local machine's interfaces to the net, in the form of a table of cumulative statistics. This display is available through `netstat` with the `-i` option.

### *netstat -i*

---

```
scobox$ netstat -i
```

| Name | Mtu  | Network    | Address       | Ipkts                   | Ierrs | Opkts | Oerrs | Collis |
|------|------|------------|---------------|-------------------------|-------|-------|-------|--------|
| en0  | 1500 | sco-eng-ne | scobox        | No Statistics Available |       |       |       |        |
| e3B0 | 1500 | 128.174.14 | 128.174.14.10 |                         | 0     | 0     | 0     | 0      |
| lo0  | 2048 | loopback   | localhost     | 189                     | 0     | 189   | 0     | 0      |

---

### *Descriptions of the display headings*

Each interface is described by a line with the following headings:

|         |                                                                                                                             |
|---------|-----------------------------------------------------------------------------------------------------------------------------|
| Name    | name of the network interface. For example, <i>en0</i> is the name of the first Ethernet interface board.                   |
| Mtu     | maximum transmission unit (in bytes). This is the largest size permitted for any single packet sent through this interface. |
| Network | name of the network address of the interface as given in <i>/etc/networks</i>                                               |

|         |                                                                                                                      |
|---------|----------------------------------------------------------------------------------------------------------------------|
| Address | name of the machine address of the interface in <i>/etc/hosts</i>                                                    |
| Ipkts   | input packets. This is the number of packets received on the interface.                                              |
| Ierrs   | input errors. This is the number of errors detected in packets of data received on this interface.                   |
| Opkts   | output packets. This is the number of packets transmitted on the interface.                                          |
| Oerrs   | output errors. This is the number of errors detected and corrected in packets of data transmitted on this interface. |
| Collis  | collisions that occurred on the network                                                                              |

## Routing tables

---

The Routing Table display provides information about the usage of each route you have configured. A route consists of a destination host or network and a network interface used to exchange packets. Direct routes are created for each interface attached to the local host.

### *netstat -r*

---

```
scobox$ netstat -r
```

```
Routing tables
```

| Destination | Gateway      | Flags | Refcnt | Use | Interface |
|-------------|--------------|-------|--------|-----|-----------|
| localhost   | localhost    | UH    | 4      | 0   | lo0       |
| sco-eng-net | scobox       | U     | 4      | 537 | en0       |
| 128.174.14  | 128.174.14.1 | U     | 0      | 0   | e3B0      |
| 128.174     | scoffle      | UG    | 0      | 0   | en0       |

```
scobox$
```

---

### ***Descriptions of the display headings***

The information displayed for each route is as follows.

|             |                                                                                                                                                                                                                                      |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Destination | network or machine to which this route allows you to connect                                                                                                                                                                         |
| Gateway     | name of the gateway you configured for this route. If you are directly connected, this is a local address. Otherwise, it is the name of the machine through which packets must be routed.                                            |
| Flags       | state of the route. Valid states are:<br><div><div>U</div><div>G</div><div>N</div><div>H</div><div>up</div><div>a route to a gateway</div><div>a route to a network</div><div>a route to a host</div></div>                          |
| Refcnt      | current number of active connections using the route. Connection-oriented protocols normally hold on to a single route for the duration of the connection, while connectionless protocols obtain a route, then discard it as needed. |
| Use         | current number of packets sent using this route                                                                                                                                                                                      |
| Interface   | name of the physical network interface used to begin the route                                                                                                                                                                       |

### ***Statistics display***

---

The Protocol Statistics display provides protocol-specific errors. The errors in the display are grouped under headings for each higher-level protocol in your system. The headings are protocol-specific.

- Internet Protocol (ip)
- Internet Control Message Protocol (icmp)
- Transmission Control Protocol (tcp)
- User Datagram Protocol (udp)

**netstat -s**

---

ip:

```
3209 total packets received
0 bad header checksums
0 with size smaller than minimum
0 with data size < data length
0 with header length < data size
0 with data length < header length
0 fragments received
0 fragments dropped (dup or out of space)
0 fragments dropped after timeout
0 packets forwarded
0 packets not forwardable
0 redirects sent
```

icmp:

```
1 call to icmp_error
0 errors not generated because old message was icmp
Output histogram:
 destination unreachable: 1
0 messages with bad code fields
0 messages < minimum length
0 bad checksums
0 messages with bad length
Input histogram:
 destination unreachable: 640
0 message responses generated
```

---

*(Continued on next page.)*



(Continued)

---

tcp:

- 348 packets sent
  - 202 data packets (3661 bytes)
  - 0 data packets (0 bytes) retransmitted
  - 101 ack-only packets (60 delayed)
  - 0 URG only packets
  - 0 window probe packets
  - 0 window update packets
  - 45 control packets
- 411 packets received
  - 233 acks (for 3654 bytes)
  - 19 duplicate acks
  - 0 acks for unsent data
  - 200 packets (1677 bytes) received in-sequence
  - 0 completely duplicate packets (0 bytes)
  - 0 packets with some dup. data (0 bytes duped)
  - 9 out-of-order packets (0 bytes)
  - 0 packets (0 bytes) of data after window
  - 0 window probes
  - 0 window update packets
  - 0 packets received after close
  - 0 discarded for bad checksums
  - 0 discarded for bad header offset fields
  - 0 discarded because packet too short
- 25 connection requests
- 12 connection accepts
- 21 connections established (including accepts)
- 72 connections closed (including 0 drops)
- 16 embryonic connections dropped
- 233 segments updated rtt (of 259 attempts)
- 0 retransmit timeouts
  - 0 connections dropped by rexmit timeout
- 0 persist timeouts
- 0 keepalive timeouts
  - 0 keepalive probes sent
  - 0 connections dropped by keepalive
- 0 connections lingered
  - 0 linger timers expired
  - 0 linger timers cancelled
  - 0 linger timers aborted by signal

udp:

- 0 incomplete headers
- 0 bad data length fields
- 0 bad checksums

---

## Chapter 3

# TCP/IP serial line communications

---

This chapter describes two Internet Protocol (IP) networking products that operate over serial lines: SLIP (Serial Line Internet Protocol) and PPP (Point-to-Point Protocol). SLIP and PPP let two computers transfer information, using the serial port in place of a network card and a standard serial cable in place of an Ethernet cable. Both SLIP and PPP are commonly used on dedicated serial links. They are useful for allowing mixes of hosts and gateways to communicate with one another (host-host, host-gateway, and gateway-gateway are all common SLIP and PPP network configurations). You can operate SLIP and PPP at any speed from 1200 baud to 19.2K baud, as long as your serial port supports the speed.

The first section of this chapter covers SLIP, and the second covers PPP. Each section describes the protocol's features, explains the configuration procedure, and offers troubleshooting suggestions.

## Administering SLIP

---

SLIP functionality is limited to defining a sequence of characters that frame IP packets (or *datagrams*) on a serial line. SLIP provides no addressing, packet type identification, or error detection or correction, and allows only asynchronous transmission. SLIP does offer a compression mechanism for packet headers. PPP, described later in this chapter, provides some enhancements to SLIP's capabilities.

The TCP/IP Runtime system includes SLIP in the basic TCP/IP installation package, so you install it automatically when you install TCP/IP.

## Configuring a SLIP connection

---

This section begins with information that you need before you begin configuring SLIP. The section then describes step-by-step procedures for configuring three types of SLIP lines: a direct SLIP connection, a dialup SLIP connection, and a SLIP/Ethernet or SLIP/Token-Ring gateway.

### *Preparing to configure a SLIP connection*

SCO TCP/IP supports up to four SLIP lines per machine. Each SLIP line requires a unique network address and hostname, making it a separate subnet. For example, a hostname for the first SLIP line on a machine called *opal* might be *opal\_slip1*, the second *opal\_slip2*, and so forth. For more information on network addresses, see the section on IP addresses in the introduction to this chapter.

The *Installation and Upgrade Guide* includes the information that you need to configure a SLIP connection. If you read this material before you configure the SLIP driver, the configuration process flows more smoothly than if you attempt to answer prompts during configuration.

Information you need to know includes:

- name of the tty line you plan to use
- baud rate at which you are operating SLIP
- IP address (or Internet address) of your system
- IP address of the target system to which you are connecting the SLIP line
- hostname of the target system to which you are connecting the SLIP line

### *Configuring a direct SLIP connection*

This section describes how to use `netconfig(ADMN)` to configure a direct SLIP connection between a machine named *emerald* and a machine named *ruby*. The following procedure assumes that the machines are connected through their COM1 serial ports using a null modem cable, and both machines have TCP/IP installed. After you enter each selected value, press `<Return>` to complete your response:

1. Boot the machine *emerald* in System Maintenance mode by entering the *root* password at the following prompt:

Type `CONTROL-D` to proceed with normal startup (or enter root passwd for System Maintenance Mode):

2. Enter the following command:

**netconfig**

3. At the Available options menu, enter the number that corresponds to Add a chain. A *chain*, in **netconfig** terms, is a communications configuration that binds networking products together as though by a chain.
4. From the list of available top-level products, enter the number that corresponds to `sco_tcp`. (SLIP is a TCP/IP product.)
5. At the prompt Select next level of chain to Add or q to quit, enter the number that corresponds to `sl0`.
6. When a prompt asks you to confirm the selected product chain, enter y if the displayed product chain is correct. If the chain is incorrect, enter n to return to the **netconfig** main menu. You can then either quit **netconfig** or add a different chain of products.
7. At the prompt Enter tty line to be used for slip, enter the name of the tty line in this format: `tty1a`.
8. At the prompt Enter Baud rate for tty1a, enter the baud rate. The default baud rate is 9600.
9. At the prompt Enter the internet address of this interface, enter the IP address for your system (*emerald*) in this format: `nnn.nnn.nnn.nnn`. See the introduction to this chapter for a discussion of IP addresses.
10. At the prompt Enter the internet address of the destination interface:, enter the address for the target system (*ruby*) in this format: `nnn.nnn.nnn.nnn`.
11. At the prompt Enter hostname of the destination interface:, enter the name of the target system; in this example, *ruby*.
12. The prompt Are these values correct? (y/n), gives you a chance to change your choices. If you wish to alter any of the values, enter n, then follow the screen prompts. If the values are correct, enter y.  
  
The **netconfig** utility automatically adds *ruby's* information to *emerald's* `/etc/hosts` file.
13. At the prompt Configure gateway?, enter n, unless you plan to establish a SLIP/Ethernet or SLIP/Token-Ring gateway as described later in this chapter.
14. When **netconfig** prompts you to add or remove pseudo ttys, it displays the current number of pseudo ttys already configured on the system. If this number is correct, enter q to quit the pseudo tty configuration screen. If you want to add or remove pseudo ttys, enter the number corresponding to the desired option and follow the prompts. For more information on pseudo ttys, see the chapter "TCP/IP network administration" earlier in this document.

15. When you again see the Available options menu, enter **q**.
16. You see a prompt to relink the kernel. Enter **y** for the first SLIP configuration. Subsequent SLIP configurations on the same machine do not require relinking the kernel.
17. During kernel relinking, enter **y** to have this kernel boot by default, and **y** again to have the kernel environment rebuilt. This procedure takes a few minutes. When the rebuild is complete, you return to the command line.
18. Disable both the modem and non-modem control ports corresponding to the tty device being used for SLIP. (The ports might already be disabled.) For example, use the following commands for COM1:  

**disable tty1a**  
**disable tty1A**
19. If you want to use header compression, edit the */etc/tcp* script, and add the **+c** flag (to turn on compression mode) or the **+e** flag (to turn on compression detection) to the **slattach(ADMN)** command. Using header compression lets you make the most efficient use of the serial line. See the **slattach(ADMN)** manual page for more information.
20. Reboot *emerald*. On booting up with the new kernel, TCP/IP should start up successfully using the SLIP interface.
21. Use the same procedure to configure the SLIP line on *ruby*, making the interface address for *ruby*'s SLIP line match the destination address for *emerald*'s SLIP line, and making the destination address for *ruby*'s SLIP line match the interface address for *emerald*'s SLIP line.

This completes the configuration of the SLIP interface on both *emerald* and *ruby*. If you want to configure more than one SLIP line on a machine, repeat the **netconfig** procedure for each line you wish to configure, selecting the option for **sl1**, and so on. Make sure that each SLIP line is a separate subnet (has a unique network address and hostname).

### *Configuring a dialup SLIP connection*

SCO TCP/IP does not directly support dialup SLIP lines. If PPP is available on both machines, configure a dialup connection using PPP rather than SLIP. However, if you cannot use PPP, you can configure dialup SLIP lines using the workaround described in this section. This workaround has not yet been tested on the current version of TCP/IP.

**NOTE** Dialup SLIP uses one of the dialer programs from the operating system's UUCP package; for example **/usr/lib/uucp/dialHA12** for a Hayes™ modem at 1200 baud. If UUCP and the appropriate dialer program are not installed, use **custom(ADM)** to install them.

The following procedure describes how to configure a dialup SLIP connection between a machine named *jade* and a machine named *diamond* across *tty1a* at 1200 baud. The procedure assumes that both machines have TCP/IP installed and SLIP configured.

1. Execute the following commands for *diamond* at the super-user shell prompt or through the */etc/tcp* script:

```
(stty 1200; echo "ATE0\r" > /dev/tty1a) < /dev/tty1a
slattach /dev/tty1a diamond_IP_address jade_IP_address 1200
```

The commands on the first line put the modem into non-echo mode, and the second line is the *slattach*(ADMN) command. *diamond* must execute the *slattach* command before *jade* connects over the dialup SLIP connection.

For information on selecting IP addresses for *diamond* and *jade*, see the introduction to this chapter.

2. Execute the following commands for *jade* at the super-user shell prompt or through a shell script:

```
(stty 1200; echo "ATE0\r" > /dev/tty1a) < /dev/tty1a
/usr/lib/uucp/dialer_program /dev/tty1a 5551212 1200

if [$?]
then
slattach /dev/tty1a jade_IP_address diamond_IP_address 1200
else
echo "Error dialing\n"
fi
```

The commands on the first line put the modem into non-echo mode, and the second line uses a UUCP dialer program to dial *diamond* at the phone number 555-1212. The subsequent lines execute *slattach* if the dialer program reports success. Make sure that *jade* dials *diamond* before issuing *slattach*.

Double-check the following points when you configure a dialup SLIP:

- each SLIP line is a separate subnet (has a unique network address and host-name).
- the *slattach* command specifies the same, supported baud rate for both machines
- both modems are using the same error protocols, data compression types, and other configurable features
- you use dedicated tty lines for the SLIP connection
- you use a non-modem control file for the device special file (for example, use */dev/tty1a* instead of */dev/tty1A* for COM1)

- the modem and non-modem control ports are disabled
- the modem is attached to the same port that the `slattach` command specifies
- the `routed(ADMN)` daemon starts after `slattach`. If `routed` starts before `slattach`, the following message appears on the console at roughly one-minute intervals:

```
routed: packet from unknown router net_address
```

If you see this message, kill `routed`, and then restart it.

### *Configuring a SLIP/Ethernet or SLIP/Token-Ring gateway*

This section explains how to set up a machine as a gateway between a SLIP network and an Ethernet network. Use the same procedure to configure a SLIP/Token-Ring gateway, a PPP/Ethernet gateway, or a PPP/Token-Ring gateway.

A SLIP/Ethernet gateway must connect two distinct networks; that is, the network address and hostname for the SLIP side must be different from those of the Ethernet side. For a complete discussion on network addresses and host addresses, see the section on IP addresses in the introduction to this chapter.

This procedure describes a SLIP/Ethernet gateway for which the machine *emerald* has the serial interface for the SLIP connection, the machine *ruby* has the Ethernet card for the Ethernet connection, and the machine *opal* is the gateway system. Both networks are configured as Class C networks. (For a discussion of network classes, see the introduction to this guide.)

This configuration procedure assumes that *opal* has TCP/IP installed, but neither the SLIP driver nor the Ethernet driver is configured:

1. Configure the SLIP interface on *opal*, using the steps outlined for a direct connection or for a dialup connection. Enter `y` at the prompt `Configure gateway?`.
2. Run `netconfig` to configure the Ethernet card on *opal*, following the instructions in the *Installation and Upgrade Guide*. Enter `y` at the prompt `Configure gateway?`.
3. Edit the `/etc/networks` file on *opal* to include the hostname and network address for each network as follows:

```
ruby_ethernet ruby_net_address
emerald_slip emerald_net_address
```

4. If `/etc/if.ignore` contains a line for `sl0`, remove the line.

5. Reboot *opal*.
6. Use `netstat -i` to check that both interfaces are configured. These three entries should appear in the first column: `loopback`, `sl0`, and an Ethernet driver such as `wdn0` or `e3B0`. Each entry should show non-zero values for incoming packets (Ipkts) and outgoing packets (Opkts).

## Removing SLIP

Remove SLIP with `netconfig`, then restore the files that you changed to reflect the SLIP line.

1. At the Available options menu, enter the number corresponding to Remove a chain, and press (Return).
2. Follow the directions in the rest of the `netconfig` screen prompts.
3. Edit the `/etc/hosts` file on each machine and remove the entry associated with each SLIP line.
4. If you are removing a SLIP/Ethernet or SLIP/Token-Ring gateway, remove the network entries from `/etc/networks`.

## Troubleshooting SLIP configurations

This section describes many of the common error messages generated from `ping(ADMN)`, `telnet(TC)`, or `rlogin(TC)`. For more information on troubleshooting your network connections, see the chapter on network administration earlier in this guide.

### Common problems with SLIP

If you encounter a problem with SLIP, check to make sure the following are true:

- `slattach` is running at both ends of the SLIP connection
- the source address of the first machine matches the destination address of the second machine, and the destination address of the first machine matches the source address of the second machine
- the machines at both ends of the connection expect to send and receive packets with compressed headers, or neither machine expects to send or receive compressed headers (that is, both machines agree about header compression)
- the machines at both ends of the connection are using the same baud rate
- the modems at both ends of the connection are using the same protocols



## *Verifying serial cable connectivity*

To verify that your problem is not a serial cable connectivity problem, follow this procedure on both machines to which the serial cable connects:

1. While logged on as *root*, bring the system to single-user mode.
2. Edit the file */usr/lib/uucp/Devices* to uncomment the following lines, by removing the number sign from each line:

```
Direct tty1a - 9600 direct
Direct tty2a - 9600 direct
```

3. After you perform the previous steps on both machines, enter the following command on the source machine: **cu -l tty1a dir**.

If the screen displays **connected**, enter some characters. They should appear on the destination machine only. If they do, your serial line is sound. Type **~** at the beginning of a line to exit the **cu** program.

If there is no response, the problem probably lies with the cable. Check the cable's connections, or try another cable. If an error message appears, see the chapter "Building a remote network with UUCP" later in this guide.

## *Troubleshooting problems with ping*

Test that the local and remote interfaces are functional and responding by booting the system in multi-user mode and executing the following **ping** commands:

```
ping local_hostname
ping remote_host_IP_address
ping remote_hostname
```

If all these commands successfully respond to transmitted packets, the connection is functional and you can try other TCP/IP commands, such as **telnet** and **rlogin**.

The following error message from **ping** indicates that the network address of the remote host is different from that of the local host:

```
ping send: Network is unreachable
```

Make sure that the network portion of the remote host's IP address is correct. (See the introduction to this chapter for a discussion of the network portion of the IP address.) For machines on the same network, the network part of the IP addresses for the local host and the remote host should match. If the machines are not on the same network, you need to configure a gateway to route packets between the two networks.

The following error message from **ping** indicates that the *remote\_hostname* used on the **ping** command line does not have an entry in */etc/hosts*:

```
ping: unknown host remote_hostname
```

Edit the */etc/hosts* file on the local host to include an entry for the remote host.

If **ping** hangs with no response:

1. Make sure that the IP address used on the command line is correct. It should be the value assigned to the machine connected to the other end of the direct or modem line.
2. Use the **ifconfig(ADMN)** command as follows to check the local SLIP interface:

```
ifconfig sln
```

This command should return an output similar to the following:

```
sln: flags=51<UP,POINTOPOINT,RUNNING>
 inet source address-->destination address netmask ffff0000
```

Check that the source and destination addresses are correct and have corresponding entries on the system connected to the other end of the line.

3. Check the netmasks of both machines. An incorrect netmask can render a host or network unreachable. For more information on netmasks, see the introduction to this chapter.

## *Troubleshooting problems with rlogin or telnet*

If **ping** works correctly but **rlogin(TC)** or **telnet(TC)** fails, consult the appropriate manual page to make sure that your command syntax is correct. If the syntax is correct, verify that your */etc/hosts* file has an entry for the target host.

The following **telnet** or **rlogin** error message indicates either that there are no more pseudo ttys available to make additional connections, or that the terminal database does not contain entries for the */dev/tty\** the system is attempting to use:

```
Cannot obtain database information on this terminal.
Connection closed.
```

For information on gaining access to additional pseudo ttys, see the section on administering pseudo ttys in the chapter "TCP/IP network administration" earlier in this guide.

## More SLIP information

---

The following manual pages provide information about SLIP:

| Manual page                 | Description                                                 |
|-----------------------------|-------------------------------------------------------------|
| <code>slip(ADMP)</code>     | SLIP implementation                                         |
| <code>slattach(ADMN)</code> | the command for assigning a tty line to a network interface |
| <code>strcf(SFF)</code>     | STREAMS configuration file format                           |

For more information about how SLIP is designed and implemented, refer to RFC 1055 by J. Romkey. For information about compressing SLIP headers, refer to RFC 1144 by Van Jacobson. You can get these documents from Jon Postel, RFC Editor, USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292-6695.

## Administering PPP

---

Some of the enhancements that Point-to-Point Protocol (PPP) provides to SLIP's capabilities include error detection, password authentication, and the ability to send data asynchronously or synchronously.

PPP's features include the following:

- dynamic connect and disconnect service to reduce phone line costs during idle time. When the phone line disconnects, the process using PPP continues in service without receiving a carrier hangup signal. When the process transmits or receives data after idling, the phone link is automatically re-established.
- serial line sharing with UUCP and `cu(C)` services
- error detection of transmitted data
- adaptability to high-speed links such as T1, cellular phones, and dial-in services
- ease of setup with the `netconfig(ADM)` administrative utility

The TCP/IP Runtime system supplies PPP as a separate installation package that you can install automatically with the rest of the TCP/IP product or separately.

**NOTE** PPP is supplied with device drivers for use with IP datagrams and for asynchronous High-level Data Link Control (HDLC) serial data transmission. PPP requires customized device drivers for synchronous transmission. Information on creating device drivers is available in the *SCO UNIX Device Driver Writer's Guide*. Note that creating a device driver is a complex and difficult task requiring specialized programming skills and a thorough knowledge of UNIX system internals.

## *Interoperability with other PPP implementations*

The following paragraphs provide configuration details for establishing PPP connections between the current implementation of PPP (in SCO Open Desktop/SCO Open Server Release 3.0 and a system running the PPP version in Open Desktop/Open Server 2.0 (TCP/IP 1.2.0) or PPP versions from other vendors.

- For connections with SCO Release 2.0 (TCP/IP Release 1.2.0) PPP:
 

On the system running SCO Open Desktop/SCO Open Server Release 3.0 PPP, include the following options in the */etc/ppphosts* file:

  - link option "accomp"
  - IP options "rfc1172addr", "ipaddr", and "VJ"
  - other option "old"
- For connections with MorningStar PPP1.3:
 

On the system running MorningStar PPP, use the "nolqm" option on the *pppd* command line. This is because the SCO Open Desktop/SCO Open Server Release 3.0 PPP implementation does not support link quality control. Also, do not use password authentication on both sides at the same time, because MorningStar PPP 1.3 only supports one side password authentication.
- For dial-up connections from FTP PC/TCP version 2.05:
 

On the system running SCO Open Desktop/SCO Open Server Release 3.0 PPP, include the options "ipaddr", "accomp", and "rfc1172addr" in the */etc/ppphosts* file. In the same file, also set *mru*=1500.

On the system running FTP PC/TCP version 2.05, set *mru*=1500, set *mtu*=1500, and turn on the "address/control field compression" flag.

- For connections with a Telebit NetBlazer 1.5:

#### Connecting to a NetBlazer from a UNIX System:

1. Configure the NetBlazer to receive the call from the UNIX system:

The NetBlazer may be set up as if the UNIX machine were just another remote 'dynamic-interface' NetBlazer, using the PPP encapsulation protocol in packet mode.

Configure the NetBlazer according to the Dynamic Interface Configuration procedure found in the NetBlazer Installation Guides, noting particularly the discussion of the `ipdial` command. When you are prompted for Name of remote(other) system:, enter the name of the UNIX system. Use PPP rather than SLIP. Set `mtu=1500`. Make a note of the dial-in user password you enter for use by the UNIX system; you will need to include this password in the UNIX system's `/usr/lib/uucp/System` entry that describes how to dial into this NetBlazer (see next step).

2. Configure the UNIX system to call the NetBlazer:

In your system's `/usr/lib/uucp/Systems` file, set the username provided by the login portion of the chat script to the UNIX system name. Make sure you set the password to the same one you entered for the dial-in user on the NetBlazer (see step 1, above). You must turn off the `ipaddr` option in the `/etc/ppphosts` file because NetBlazer rejects ip-address negotiation.

#### Connecting to a UNIX system from a NetBlazer:

1. Configure the UNIX system to receive the call from NetBlazer:

Turn off the `ipaddr` option in the `/etc/ppphosts` file because NetBlazer 1.5 rejects ip-address negotiation.

## 2. Configure the NetBlazer to call the UNIX system:

Add the following entry to the NetBlazer's CHAT.TXT file:

```
for connecting to a SCO UNIX machine.
:unix
s1,20,30,2,
e2,20,30,3,in:
e3,3,4,4,$6%~&
u4,5,100,5
s5,5,100,6,
e6,20,30,7,word:
e7,3,8,8,$6%~&
p8,5,100,9
s9,5,100,99,

s30,20,30,31,
e31,20,30,32,in:
e32,3,33,33,$6%~&
u33,5,100,34
s34,5,100,35,
e35,20,100,36,word:
e36,3,37,37,$6%~&
p37,5,100,38
s38,5,100,99,

r99,0
```

Reboot the NetBlazer to effect any changes to the CHAT.TXT file. When you configure the NetBlazer's dynamic interface to call the UNIX system (as described in "Configure the NetBlazer to receive the call from the UNIX system", above), specify unix for the chat script instead of the default ics.

## PPP compared to SLIP

PPP was created to enhance the capabilities of the Serial Line Internet Protocol (SLIP) discussed earlier in this chapter. The following table compares these two products:

| SLIP                            | PPP                                                            |
|---------------------------------|----------------------------------------------------------------|
| Asynchronous transmission only  | Asynchronous or synchronous                                    |
| Internet Protocol only          | Multiple protocol stack capability                             |
| Serial line always in use       | Dynamic connect and disconnect optimizes line use during idles |
| Exclusive line use              | Can be shared with UUCP and cu                                 |
| No error detection of datagrams | Built-in error detection                                       |

## Configuring PPP

---

This section begins with information that you need before you begin configuring PPP, then describes the **netconfig** procedure for configuring PPP. Finally, the section explains how to edit the *UUCP Systems* and *Devices* files (and the *Dialers* file, if the PPP line operates over modems) to configure the software for a PPP serial line.

You can edit the *Systems*, *Devices*, and *Dialers* files before or after running **netconfig**. Editing them before running **netconfig** is better, so that you can verify connectivity.

### *Preparing to configure PPP*

Before configuring PPP, make sure that UUCP is installed on your system. Refer to the chapter “Building a remote network with UUCP” later in this guide for information about how to set up UUCP.

**netconfig** supports up to four PPP lines per machine. Each PPP line requires a unique network address and hostname, making it a separate subnet. For example, a hostname for the first PPP line on a machine called *opal* might be *opal\_ppp1*, the second *opal\_ppp2*, and so forth. Network addresses are discussed in the section on IP addresses in the introduction to this guide.

The *Installation and Upgrade Guide* contains the information that you need to configure a PPP connection. If you read this material before you configure the SLIP driver, the configuration process flows more smoothly than if you attempt to answer prompts during configuration.

Information you need to know includes:

- IP address (or Internet address) of your system
- IP address of the target system to which the PPP line is being connected
- netmask for this interface
- remote system’s ID and password if you want to use the Password Authentication Protocol. For more information, see the online reference manual page *pppauth*(SFF).

For more information on IP addresses and netmasks, see the introduction to this chapter.

### *Configuring PPP with netconfig*

This section describes how to use **netconfig**(ADMN) to configure PPP. The **netconfig** utility can configure several PPP serial lines. Even though PPP permits dynamic IP address negotiation, **netconfig** treats each PPP connection as

though it were a SLIP line and requires explicit entry of IP addresses for your system and for the target system. You can consider the target system address a starting address with negotiation available when you use PPP.

**NOTE** During configuration, the **netconfig** utility supplies information to the */etc/hosts*, */etc/strcf*, and */etc/tcp* files. None of these files require additional editing during configuration.

The following procedure describes how to configure PPP between a machine named *emerald* and a machine named *ruby*. After you enter each selected value, press the (Return) key to complete your response:

1. Boot the machine *emerald* in System Maintenance mode by entering the *root* password at the following prompt:

Type CONTROL-D to proceed with normal startup (or enter root  
passwd for System Maintenance Mode):

2. Enter the following command and press (Return):  
**netconfig**
3. At the Available options menu, enter the number that corresponds to Add a chain. A *chain*, in **netconfig** terms, is a communications configuration that binds networking products together as though by a chain.
4. From the list of available top-level products, enter the number that corresponds to *sco\_tcp*.
5. At the prompt Select next level of chain to Add or q to quit:, enter the number that corresponds to *ppp0*.
6. When a prompt asks you to confirm the selected product chain, enter *y* if the displayed product chain is correct. If the chain is incorrect, enter *n* to return to the **netconfig** main menu. You can then either quit **netconfig** or add a different chain of products.
7. The first time that you configure a PPP connection, follow the **netconfig** prompts to choose a password.
8. At the prompt Enter the internet address of this interface:, enter the IP address for your system (*emerald*) in this format: *nnn.nnn.nnn.nnn*.
9. At the prompt Enter the netmask for this interface, enter the netmask in this format: *nnn.nnn.nnn.nnn*.
10. At the prompt Enter the internet address of the destination interface:, enter the IP address for the target system (*ruby*) in this format: *nnn.nnn.nnn.nnn*.



11. At the prompt Enter hostname of the destination interface:, enter the name of the target system; in this example, **ruby**.
12. The prompt Are these values correct? (y/n) gives you a chance to change your choices. If you want to alter any of the values, enter **n**, then follow the screen prompts. If the values are correct, enter **y**.
13. At the prompt Enter local host name or enter q to quit [ID]:, enter the name of this system; in this example, **emerald**. This name will be used in the "peer\_id" field of a sent PAP Authenticate\_Request message.
14. At the prompt Enter local host password for PPP authentication or enter q to quit [PASSWD]:, enter the password that the peer also knows. The password will be used in the "password" field of a sent PAP Authenticate\_Request message.
15. At the prompt Configure gateway?, enter **n**, unless you plan to establish a PPP/Ethernet or PPP/Token-Ring gateway as described in the SLIP section of this chapter. (You only see this prompt if ppp0 is the first driver configured after the loopback driver.)
16. When **netconfig** prompts you to add or remove pseudo ttys, enter **q** if you want to maintain the pseudo ttys currently configured. If you want to change the number of pseudo ttys, enter **y**, and consult the section on administering pseudo ttys in the chapter "TCP/IP network administration" earlier in this document.
17. When **netconfig** prompts you to add or remove TCP connections, enter **q** if you want to maintain the TCP connections currently configured. If you want to change the number of TCP connections, enter **y**.
18. When you again see the Available options: menu, enter **q**.
19. You see a prompt to relink the kernel. Enter **y** for the first PPP configuration. Subsequent PPP configurations on the same machine do not require relinking the kernel.
20. During kernel relinking, enter **y** to have this kernel boot by default, and **y** again to have the kernel environment rebuilt. This procedure takes a few minutes. When the rebuild is complete, you return to the command line.
21. Reboot *emerald*.
22. Enable the tty line that PPP is using.
23. Make sure that */etc/ppphosts* has the correct entry in the tty column for the chosen connection (modem or direct). The correct entry for a modem connection is a dash (-); the correct entry for a direct connection is the name of the device that PPP is using (for example, */dev/tty1a*).

24. Use the same procedure to configure the PPP line on *ruby*, making the interface address for *ruby*'s PPP line match the destination address for *emerald*'s PPP line, and making the destination address for *ruby*'s PPP line match the interface address for *emerald*'s PPP line.

If you want to configure more than one PPP line on a machine, repeat the **netconfig** procedure for each line you wish to configure, selecting the option for **ppp1**, and so on. Make sure that each PPP line is a separate subnet (has a unique network address and hostname).

## Adding PPP information to configuration files

The next step in the configuration procedure is to add information to the parameter files used by UUCP and PPP. If your PPP line operates over modems, you might also need to add information to a file that controls the dial activity. Each file has an associated manual page as follows:

| File                  | Description        | Manual page | Permissions |
|-----------------------|--------------------|-------------|-------------|
| /usr/lib/uucp/Systems | remote system info | systems(F)  | 600         |
| /usr/lib/uucp/Devices | device info        | devices(F)  | 644         |
| /usr/lib/uucp/Dialers | dialing info       | dialers(F)  | 644         |

You can edit these files directly, using your choice of text editor, or you can run **uucinstall**(ADM), which prompts you for the necessary information.

**NOTE** If you edit the parameter files directly, create a backup copy of each file and refer to the respective manual page before you make any changes. When you finish editing a file, make sure that the permissions modes are the same as those shown for each file. This is especially important for the *Systems* file, which contains high-security information.

The remainder of this section briefly presents information and examples to help you edit the parameter files either directly or through **uucinstall**. For more details about these files, see the UUCP chapter in your administrator's guide.

Make sure that site names are consistently named in each file.

*/usr/lib/uucp/Systems:*

```
pppdir Any Direct 2400-9600 - "" \r ogin:--ogin: nppp word: testing
pppmodem Any ACU 9600 5555 "" \r ogin:--ogin: nppp word: testing
```

The first entry is for a PPP direct line called **pppdir** that can operate at speeds from 2400 to 9600 baud. The second entry is for a modem line that uses an Automatic Calling Unit (ACU) that operates at 9600 baud. This dialer is accessed through an internal phone system, such as a PBX, and is reached at extension 5555. Both lines have **nppp** as their login keyword.

#### */usr/lib/uucp/Devices:*

```
Direct ttyla - 2400-9600 direct
ACU ttylA - 9600 atdialUSR
```

For this file, the UUCP installation might supply sufficient entries. Make sure that the file provides direct information, as the first entry shows. If PPP is used with a modem, make sure that the file contains a line such as the second entry.

#### */usr/lib/uucp/Dialers:*

This file might already contain the appropriate entries. Refer to the **dialers(F)** manual page and the chapter "Using Modems" later in this section for instructions on adding information to the *Dialers* file.

### ***Configuring a PPP/Ethernet or PPP/Token-Ring gateway***

To set up a machine as a gateway between a PPP network and an Ethernet network, follow the procedure for configuring a SLIP/Ethernet gateway that appears earlier in this chapter. Configuring a gateway machine between a PPP network and a Token-Ring network also uses the same procedure.

### ***Removing PPP***

Remove PPP with **netconfig**, and then restore the files that you changed to reflect the PPP line.

1. At the Available options prompt, enter the number corresponding to Remove a chain, and press (Return).
2. Follow the directions in the rest of the **netconfig** screen prompts, entering **y** to relink the kernel.
3. If you are removing a PPP/Ethernet or PPP/Token-Ring gateway, remove the network entries from */etc/networks*.
4. If you are removing your last link to another machine, remove the entry for that machine from your *Systems* file for security reasons. You can use **uinstall** to edit the *Systems* file.

### ***Troubleshooting PPP***

---

To begin troubleshooting PPP, prepare a test environment as follows:

1. Place two computers side-by-side and connect with a direct serial line. Generally, a null modem cable should cross-connect the serial ports between the two computers.
2. Make sure that the PPP software is installed correctly by running **netconfig** and verifying that the display matches your expectations.

3. Be familiar with the "Troubleshooting your system" chapter in the *Administering Operating System Services* section of this book.
4. Bring both systems to a known state. For example, reboot both systems, and bring them to multiuser mode.
5. Using multiscreens, monitor the system message log on each system with:

```
tail -f /usr/adm/syslog
```

The following guidelines can help you isolate the problem and find the solution:

#### **Does the serial line function?**

For a PPP connection on *tty1a* running at 1200 baud, enter the following command on each machine:

```
cu -ltty1a -s1200 dir
```

This command should cause characters typed on one machine to display on the other. If not, make sure that the */usr/lib/uucp* file is correct, and make sure that your cable does not have a short. If the *cu* command does display the characters on the other machine, then run *netconfig* to make sure that the PPP is installed. If it is installed, then make sure that a device node exists for the *tty*, and that the serial cable is securely connected.

#### **Does the modem connection work?**

For a PPP connection on *tty1a* running at 1200 baud and calling a modem at extension 5555, enter the following command on each machine:

```
cu -ltty1a -s1200 5555
```

The modem should dial, connect, and present a login prompt. If you do not see a login prompt after many seconds, type *~%B* to make sure that the remote modem did not accidentally cycle past the correct baud rate. Try *~%B* three times, waiting several seconds in between each try. If you still do not get a login prompt, make sure that the entries in your *Devices* file are correct, and that the remote *tty* setup is correct.

Next, enter the following command on each machine, specifying the remote hostname:

```
cu hostname
```

You should get the same results as when you specify the extension number. If you do not, make sure that the entries in your *Systems* are correct.

#### Does ping work?

Follow the directions in the SLIP troubleshooting section earlier in this chapter to test the connection with the **ping** command. If **ping** does not work, make sure that the hostname listed in */etc/ppphosts* is correct.

#### Does telnet work?

If you run **telnet** or one of the other utilities for using PPP, and messages do not appear or the command fails, check the */usr/adm/syslog* display. If this does not provide the answer, try disabling and enabling the tty device associated with the PPP serial line. You can also check that the **uutry(ADM)** and **uustat(C)** utilities work. Follow the directions in the Performance and Troubleshooting section of this book. This chapter has detailed directions for handling error messages such as **DEVICE LOCKED**, **LOGIN FAILED**, or **NO DEVICES AVAILABLE**.

For additional troubleshooting ideas, consult the section on troubleshooting SLIP lines that appears earlier in this chapter.

## More PPP information

---

The following manual pages provide information about PPP:

| Manual page           | Description                                         |
|-----------------------|-----------------------------------------------------|
| <b>asyhdlc(ADMP)</b>  | asynchronous HDLC device driver                     |
| <b>ifconfig(ADMN)</b> | configure network parameters                        |
| <b>if.ignore(SFF)</b> | specifies which daemons should not be sent over PPP |
| <b>ppp(ADMN)</b>      | PPP login shell                                     |
| <b>ppp(ADMP)</b>      | PPP implementation                                  |
| <b>pppd(ADMN)</b>     | PPP daemon                                          |
| <b>ppphosts(SFF)</b>  | PPP host attributes                                 |
| <b>pppauth(SFF)</b>   | PPP authentication database                         |
| <b>strcf(SFF)</b>     | STREAMS configuration file format                   |

For more information about how PPP is designed and implemented, refer to RFC 1331, RFC 1332 and the Internet Draft "The PPP Authentication Protocols" (the Draft is by B. Lloyd and W. Simpson). See the online **ppp(ADMP)** manual page for a description of what is not implemented from these documents. These documents are available from the InterNIC Registration Services in Herndon, Virginia.

See "Registering domain names" (page 723) for complete information on contacting the InterNIC.

## Chapter 4

# Configuring the BIND name server

---

With the Berkeley Internet Name Domain (BIND) server, you can create and maintain a distributed host name and address database for computers on your network. Your system is configured by default to use the network hosts file */etc/hosts* found on each computer. If you have a large network, updating every computer's */etc/hosts* file can be time-consuming. Using BIND frees the system administrator from continually updating host name and address data on every machine.

This chapter explains basic BIND concepts, describes BIND configuration tasks, and provides examples of the various BIND configuration files.

## The name service

---

The basic function of the name server is to answer queries from clients about host names and addresses. With the name server, the network is broken into a hierarchy of domains. The name space is organized as a tree, according to organizational or administrative boundaries. Each node, called a domain, is given a label, and the name of the domain is the concatenation of all the labels of the domains from the *root* to the current domain, listed from right to left, separated by dots. A label need only be unique within its domain. The whole space is partitioned into several areas called zones, each starting at a domain and extending down to the leaf domains or to domains where other zones start. Zones usually represent administrative boundaries. An example of a host address for a host at the University of California, Berkeley, would look as follows:

monet . Berkeley . EDU

The top-level domain for educational organizations is EDU; Berkeley is a sub-domain of EDU and monet is the name of the host. Additional top-level domains include:

COM commercial organizations

GOV government organizations

MIL military departments

ORG miscellaneous organizations

## *Types of servers*

---

There are several types of servers. These are:

- master servers
- caching-only servers
- remote servers
- slave servers

These types of servers are described in more detail in the following four sections.

### *Master servers*

---

A “master” server for a domain is the authority for that domain. This server maintains all the data corresponding to its domain. Each domain should have at least two master servers: a primary master, and some secondary masters to provide backup service if the primary is unavailable or overloaded. A server may be a master for multiple domains, being primary for some domains and secondary for others.

#### *Primary*

A “primary master” server is a server that loads its data from a file on disk. This server may also delegate authority to other servers in its domain.

#### *Secondary*

A “secondary master” server is a server that is delegated authority and receives its data for a domain from a primary master server. At boot time, the secondary server requests all the data for the given zone from the primary master server. This server then periodically checks with the primary server to see if it needs to update its data.

## Caching-only servers

---

All servers are “caching” servers. This means that the server caches the information that it receives for use until the data expires. A caching only server is a server that is not authoritative for any domain. This server services queries and asks other servers that have the authority for the information needed. All servers keep data in their caches until the data expires, based on a time-to-live field attached to the data when it is received from another server.

## Remote servers

---

A “remote” server is an option given to people who would like to use a name server on their workstation or on a machine that has a limited amount of memory and CPU cycles. With this option, you can run all of the networking programs that use the name server without running the name server on the local machine. All of the queries are serviced by a name server that is running on another machine on the network.

## Slave server

---

A “slave” server is a server that always forwards queries it cannot satisfy locally to a fixed list of forwarding servers instead of interacting with the master name servers for the *root* and other domains. The queries to the forwarding servers are recursive queries. There may be one or more forwarding servers, and they are tried in turn until the list is exhausted. A slave and forwarder configuration is typically used when you do not wish all the servers at a given site to be interacting with the rest of the Internet servers.

A typical scenario would involve a number of workstations and a departmental timesharing machine with Internet access. The workstations might be administratively prohibited from having Internet access. To give the workstations the appearance of access to the Internet domain system, the workstations could be slave servers to the timesharing machine, which would forward the queries and interact with other name servers to resolve the query before returning the answer. An added benefit of using the forwarding feature is that the central machine develops a much more complete cache of information that all the workstations can take advantage of. The use of slave mode and forwarding is discussed further under the description of the *named* boot file commands.



## Setting up your own domain

---

When setting up a domain that is going to be on a public network, the site administrator should contact the organization in charge of the network and request the appropriate domain registration form.

**NOTE** If you ever want to connect to a public network, you should reserve a name before customizing your name server files. Doing so ensures that you do not choose a name that is already in use elsewhere on the network.

An organization that belongs to multiple networks (such as the Internet and BITNET) should register with only one network.

The contacts are as follows:

### *Internet*

Sites that are already on the Internet and need information on setting up a domain should contact `hostmaster@internic.net`. You may also want to be placed on the BIND mailing list, which is a mail group for people on the Internet running BIND. This group discusses future design decisions, operational problems, and other related topics. To request placement on this mailing list, send mail to the following address:

`bind-request@ucbarpa.Berkeley.EDU`.

If you have access to news over the network, you can also subscribe to the newsgroup `comp.protocols.tcp-ip.domains` for more discussion of domains.

### *BITNET*

If you are on the BITNET and need to set up a domain, contact `INFO@BITNIC`.

## Boot file

---

The name server uses several files to load its database. The major file used is the boot file. This is the file that is first read when `named` starts up. This tells the server what type of server it is, which zones it has authority over, and where to get its initial data. The default location for this file is `/etc/named.boot`. However, this can be changed by setting the `BOOTFILE` variable or by specifying the location on the command line when `named` starts up.

### *Directory*

The directory line specifies the directory in which the name server should run, allowing the other filenames in the boot file to use relative pathnames.

directory            `/usr/local/lib/named`

If you have more than a couple of named files to be maintained, you may wish to place the named files in a directory such as `/usr/local/domain` and adjust

the directory command properly. The main purposes of this command are to make sure **named** is in the proper directory when trying to include files by relative pathnames with **\$INCLUDE** and to allow **named** to run in a location that is reasonable to dump core.

### *Primary master*

The line in the boot file that designates the server as a primary server for a zone looks like the following:

```
primary Berkeley . Edu ucbhosts
```

The first field specifies that the server is a primary one for the zone stated in the second field. The third field is the name of the file from which the data is read.

### *Secondary master*

The line for a secondary server is similar to that for the primary, except that it lists addresses of other servers (usually primary servers) from which the zone data is obtained.

```
secondary Berkeley . Edu 128.32.0.10 128.32.0.4
```

The first field specifies that the server is a secondary master server for the zone stated in the second field. The two network addresses specify the name servers that are primary for the zone. The secondary server gets its data across the network from the listed servers. Each server is tried in the order listed until it successfully receives the data from a listed server. If a filename is present after the list of primary servers, data for the zone is dumped into that file as a backup. When the server is first started, the data are loaded from the backup file if possible, and a primary server is then consulted to check that the zone is still up-to-date.

### *Caching-only server*

You do not need a special line to designate that a server is a caching server. A caching-only server is indicated by the absence of authority lines, such as secondary or primary in the boot file.

All servers should have the following line in the boot file to prime the name server's cache:

```
cache . root.cache
```

The period "." specifies the domain. All cache files listed are read in at named boot time and any values still valid are reinstated in the cache and the *root* name server information in the cache files are always used. For information on the cache file, see the section "Initializing the Cache" later in this chapter.

### *Forwarders*

Any server can make use of forwarders. A forwarder is another server capable of processing recursive queries to try to resolve queries on behalf of other systems.

The **forwarders** command specifies forwarders by Internet address as follows:

```
forwarders 128.32.0.10 128.32.0.4
```

There are two main reasons for wanting to do so. First, the other systems may not have full network access and may be prevented from sending any IP packets into the rest of the network and, therefore, must rely on a forwarder that does have access to the full net. The second reason is that the forwarder sees a union of all queries as they pass through the forwarder's server and, therefore, the forwarder builds up a very rich cache of data compared to the cache in a typical workstation name server. In effect, the forwarder becomes a meta-cache that all hosts can benefit from, thereby reducing the total number of queries from that site to the rest of the net.

### *Slave mode*

Slave mode is used if the use of forwarders is the only possible way to resolve queries because of lack of full net access or if you wish to prevent the name server from using other than the listed forwarders. Slave mode is activated by placing the simple command in the boot file:

```
slave
```

If **slave** is used, then you must specify forwarders. When in slave mode, the server forwards each query to each of the forwarders until an answer is found or the list of forwarders is exhausted.

## *Remote servers*

---

Every remote server on your system requires the file */etc/resolv.conf*. This file designates the name servers on the network that should be queried.

## *Initializing the cache*

---

The name server needs to know the identities of the authoritative name servers for the *root* domain of the network. To do this, you have to prime the name server's cache with the address of these higher authorities. This is done in a file called *root.cache*. The location of this file is specified in the boot file */etc/named.boot*.

## Standard files

---

There are three standard files used to specify the data for a domain. These files are:

*named.local*  
*named.hosts*  
*named.rev.*

The *named.local* file specifies the address for the local loopback interface, better known as *localhost*, with the network address 127.0.0.1. The location of this file is specified in the boot file.

The *named.hosts* file contains all the data about the machines in this zone. The location of this file is specified in the boot file.

The *named.rev* file specifies the IN-ADDR.ARPA domain. This is a special domain for allowing address-to-name mapping. Because Internet host addresses do not fall within domain boundaries, this special domain was formed to allow inverse mapping. The IN-ADDR.ARPA domain has four labels preceding it. These labels correspond to the four octets of an Internet address. All four octets must be specified even if an octet is zero. The Internet address 128.32.0.4 is located in the domain 4.0.32.128.IN-ADDR.ARPA. This reversal of the address is awkward to read but allows for the natural grouping of hosts in a network.

## Standard resource records

---

The records in the name server data files are called resource records. The following is a general description of a resource record:

| (name) | {ttl} | addr-class | Record Type | Record Specific data |
|--------|-------|------------|-------------|----------------------|
|--------|-------|------------|-------------|----------------------|

Resource records have a standard format, as shown above. The first field is always the name of the domain record and it must always start in column 1. For some resource records, the name can be left blank. In such cases, the name of the previous resource record is used. The second field is an optional time-to-live field. This specifies how long this data is stored in the database. When this field is left blank, the default time-to-live is specified in the Start of Authority resource record discussed later in this chapter. The third field is the address class. There are currently two classes: IN for Internet addresses and ANY for all address classes. The fourth field states the type of the resource record. The fields after that are dependent on the type of the resource record. Case is preserved in names and data fields when loaded into the name server. All comparisons and lookups in the name server database are case-insensitive.

#### 4. Configuring the BIND name server

The following characters have special meanings:

- `.` A free-standing dot in the name field refers to the current domain.
- `"@"` A free-standing "@" in the name field denotes the current origin. You can use origins within configuration files as a shorthand for fully qualified domain names.
- `..` Two free-standing dots represent the null domain name of the *root* when used in the name field.
- `\X` Here *X* is any character other than a digit (0-9), and `\X` escapes that character so that its special meaning does not apply. For example, `\.` can be used to place a dot character in a label.
- `\DDD` Here each *D* is a digit, and `\DDD` is the octet corresponding to the decimal number described by *DDD*. The resulting octet is assumed to be text and is not checked for special meaning.
- `()` Parentheses are used to group data that crosses a line. In effect, line terminations are not recognized within parentheses.
- `;` A semicolon starts a comment; the remainder of the line is ignored.
- `*` An asterisk signifies a wildcard.

Most resource records have the current origin appended to names if they are not terminated by a `"."`. This is useful for appending the current domain name to the data, such as machine names, but can cause problems when you do not want this to happen. The following is a good rule of thumb: if the name is not in the domain for which you are creating the data file, end the name with a `"."`.

### *Separating data into multiple files*

---

An include line begins with `$INCLUDE` (starting in column 1) and is followed by a filename. This feature is particularly useful for separating different types of data into multiple files. Here is an example:

```
$INCLUDE /usr/named/data/mailboxes
```

The line would be interpreted as a request to load the file `/usr/named/data/mailboxes`. The `$INCLUDE` command does not cause data to be loaded into a different zone or tree. This is simply a way to allow data for a given zone to be organized in separate files. For example, mailbox data might be kept separately from host data using this mechanism.

## Changing an origin in a data file

---

Use the \$ORIGIN command to change the origin in a data file. The line starts in column 1 and is followed by a domain origin. This is useful for putting more than one domain in a data file. For example, */etc/named.hosts* might contain lines of the form:

```
$ORIGIN CC.Berkeley.EDU
[assorted domain data...]
$ORIGIN EE.Berkeley.EDU
[assorted domain data...]
```

## The start of authority resource record (SOA)

---

The Start of Authority record designates the start of a zone. An SOA record includes the following fields:

|                  |                                                                                                                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name             | name of the zone                                                                                                                                                                                |
| Origin           | name of the host on which this data file resides                                                                                                                                                |
| Person in charge | mailing address for the person responsible for the name server                                                                                                                                  |
| Serial number    | version number of this data file; this number should be incremented whenever a change is made to the data. (Note that the name server cannot handle numbers over 9999 after the decimal point.) |
| Refresh          | how often, in seconds, a secondary name server is to check with the primary name server to see if an update is needed                                                                           |
| Retry            | how long, in seconds, a secondary server is to retry after a failure to check for a refresh                                                                                                     |
| Expire           | upper time limit, in seconds, that a secondary name server is to use the data before it expires for lack of getting a refresh                                                                   |
| Minimum          | default number of seconds to be used for the time-to-live field on resource records                                                                                                             |

There should only be one SOA record per zone.

#### 4. Configuring the BIND name server

Here is an example of an SOA record:

| name (ttl) | addr-class | SOA     | Origin               | Person in charge           |
|------------|------------|---------|----------------------|----------------------------|
| @          | IN         | SOA     | ucbvax.Berkeley.Edu. | kjd.ucbvax.Berkeley.Edu. ( |
|            |            | 1.1     | ; Serial             |                            |
|            |            | 3600    | ; Refresh            |                            |
|            |            | 300     | ; Retry              |                            |
|            |            | 3600000 | ; Expire             |                            |
|            |            | 3600)   | ; Minimum            |                            |

#### *The name server resource record (NS)*

---

The name server record (NS) lists a name server responsible for a given domain. The first name field lists the domain that is serviced by the listed name server. There should be one NS record for each master server for the domain. Here is an example of a name server record:

| (name) | (ttl) | addr-class | NS | Name servers name         |
|--------|-------|------------|----|---------------------------|
| @      |       | IN         | NS | ucbarpa . Berkeley . Edu. |

The "@" character specifies the current origin. The address class is IN (Internet addresses), and the record type is name server (NS). The record uses the default ttl (time-to-live) value. Here, the record-specific data is the identity of the name server.

#### *The address resource record (A)*

---

The address record (A) lists the address for a given machine. The name field is the machine name and the address is the network address. There should be one A record for each address of the machine. Here is an example of an address record for a machine named *ucbarpa* with two network addresses:

| (name)  | (ttl) | addr-class | A | address    |
|---------|-------|------------|---|------------|
| ucbarpa |       | IN         | A | 128.32.0.4 |
|         |       | IN         | A | 10.0.0.78  |

#### *The host information resource record (HINFO)*

---

The host information resource record (HINFO) is for host-specific data. It lists the hardware and operating system that are running at the listed host. It should be noted that only a single space separates the hardware information and the operating-system information. If you want to include a space in the machine name, you must quote the name. Host information is not specific to any address class, so ANY may be used for the address class. There should be one HINFO record for each host. Here is an example:

| (name) | (ttl) | addr-class | HINFO | Hardware   | OS   |
|--------|-------|------------|-------|------------|------|
|        |       | ANY        | HINFO | VAX-11/780 | UNIX |

Note that the current release ignores any records that appear after an HINFO record. Thus, you can use only one HINFO record within the file, and it should be the last record in the file.

## *The well-known services resource record (WKS)*

The well-known services record (WKS) describes the well-known services supported by a particular protocol at a specified address. The list of services and port numbers comes from the list of services specified in */etc/services*. There should be only one WKS record per protocol per address. Here is an example of a WKS record:

```
(name) (ttl) addr-class WKS address protocol list of services
 IN WKS 128.32.0.10 UDP who route timed domain
 IN WKS 128.32.0.10 TCP (echo telnet
discard sunrpc sftp
uucp-path systat daytime
netstat qotd nntp
link chargen ftp
auth time whois mtp
pop rje finger smtp
supdup hostnames
domain
name server)
```

## *The canonical name resource record (CNAME)*

The canonical name resource record (CNAME) specifies an alias for a canonical name. An alias should be the only record associated with the alias name; all other resource records should be associated with the canonical name and not with the alias. Any resource records that include a domain name as their value (for example, NS or MX) should list the canonical name, not the alias. Here is an example of a CNAME record:

```
aliases (ttl) addr-class CNAME Canonical name
ucbmonet IN CNAME monet
```

## *The domain name pointer resource record (PTR)*

A domain name pointer record (PTR) allows special names to point to some other location in the domain. The following example of a PTR record is used in setting up reverse pointers for the special IN-ADDR.ARPA domain. This line is from the example:

```
named.rev file.
```

In this record, the name field is the network number of the host in reverse order. You only need to specify enough octets to make the name unique. For example, if all hosts are on network 127.174.14, then only the last octet needs



#### 4. Configuring the BIND name server

to be specified. If hosts are on networks 128.174.14 and 127.174.23, then the last two octets need to be specified. PTR names should be unique to the zone. Here is an example of a PTR record:

| name | {ttl} | addr-class | PTR | real name                |
|------|-------|------------|-----|--------------------------|
| 7.0  |       | IN         | PTR | monet . Berkeley . Edu . |

#### *The mailbox resource record (MB)*

---

The mailbox resource record has a record type of MB. It lists the machine where a user wants to receive mail. The name field is the user's login; the machine field denotes the machine to which mail is to be delivered. Mail box names should be unique to the zone. Here is an example of an MB record:

| name   | {ttl} | addr-class | MB | Machine         |
|--------|-------|------------|----|-----------------|
| miriam |       | IN         | MB | vineyd.DEC.COM. |

#### *The mail rename resource record (MR)*

---

The mail rename record (MR) lists aliases for a user. The name field lists the alias for the name listed in the fourth field, which should have a corresponding MB record. Here is an example of a mail rename record:

| name         | {ttl} | addr-class | MR | corresponding MB |
|--------------|-------|------------|----|------------------|
| Postmistress |       | IN         | MR | miriam           |

#### *The mailbox information resource record (MINFO)*

---

The mail information record MINFO creates a mail group for a mailing list. This resource record is usually associated with a mail group, but it can be used with a mail box record. The "name" specifies the name of the mailbox. The "requests" field is where mail such as requests to be added to a mail group should be sent. The "maintainer" is a mailbox that should receive error messages. This is particularly appropriate for mailing lists when errors in members' names should be reported to a person other than the sender. Here is an example of this record:

| name | {ttl} | addr-class | MINFO | requests     | maintainer        |
|------|-------|------------|-------|--------------|-------------------|
| BIND |       | IN         | MINFO | BIND-REQUEST | kjd.Berkeley.Edu. |

#### *The mail group member resource record (MG)*

---

The mail group record (MG) lists members of a mail group.

| (mail group name) | {ttl} | addr-class | MG | member name |
|-------------------|-------|------------|----|-------------|
|                   |       | IN         | MG | Bloom       |

An example for setting up a mailing list is as follows:

|      |    |       |                       |                   |
|------|----|-------|-----------------------|-------------------|
| Bind | IN | MINFO | Bind-Request          | kjd.Berkeley.Edu. |
|      | IN | MG    | Ralph.Berkeley.Edu.   |                   |
|      | IN | MG    | Zhou.Berkeley.Edu.    |                   |
|      | IN | MG    | Painter.Berkeley.Edu. |                   |
|      | IN | MG    | Riggle.Berkeley.Edu.  |                   |
|      | IN | MG    | Terry.pa.Xerox.Com.   |                   |

## *The mail exchanger resource record (MX)*

---

| name           | {ttl} | addr-class | MX | preference | value | mailer          | exchanger |
|----------------|-------|------------|----|------------|-------|-----------------|-----------|
| Munnari.OZ.AU. |       | IN         | MX | 0          |       | Seismo.CSS.GOV. |           |
| *.IL.          |       | IN         | MX | 0          |       | RELAY.CS.NET.   |           |

Mail exchanger records (MX) identify a machine that knows how to deliver mail to a machine that is not directly connected to the network. In the first example above, Seismo.CSS.GOV. is a mail gateway that knows how to deliver mail to Munnari.OZ.AU. but other machines on the network cannot deliver mail directly to Munnari. These two machines may have a private connection or use a different transport medium. The preference value is the order that a mailer should follow when there is more than one way to deliver mail to a single machine. See RFC974 for more detailed information.

Wildcard names containing the character "\*" may be used for mail routing with MX records. There are likely to be servers on the network that simply state that any mail to a domain is to be routed through a relay. In the second example above, all mail to hosts in the domain IL is routed through RELAY.CS.NET. This is done by creating a wildcard resource record, which states that \*.IL has an MX of RELAY.CS.NET.

## *Some sample files*

---

The following sections contain sample files for the name server. This covers example boot files for the different types of server and example domain database files.

### *Caching-only server*

---

```
;
; Boot file for Caching Only Name Server
;
; type domain source file or host
;
domain Berkeley.Edu
cache . /etc/named.ca
primary 0.0.127.in-addr.arpa /etc/named.local
```

### *Primary master server*

---

```
;
; Boot file for Primary Master Name Server
;
; type domain source file or host
;
directory /usr/local/lib/named
primary Berkeley.Edu ucbhosts
primary 32.128.in-addr.arpa ucbhosts.rev
primary 0.0.127.in-addr.arpa named.local
cache . root.cache
```

### *Secondary master server*

---

```
;
; Boot file for Secondary Name Server
;
; type domain source file or host
;
directory /usr/local/lib/named
secondary Berkeley.Edu 128.32.0.4 128.32.0.10 128.32.136.22 ucbhosts.bak
secondary 32.128.in-addr.arpa 128.32.0.4 128.32.0.10 128.32.136.22 ucbhosts.rev.bak
primary 0.0.127.in-addr.arpa named.local
cache root.cache
```

### *The `letchresolv.conf` file*

---

```
domain Berkeley.Edu
nameserver 128.32.0.4
nameserver 128.32.0.10
```

## *root.cache*

The addresses shown in this file are for example only, and they do not indicate valid addresses. You need to edit your own *root.cache* file to contain valid addresses.

```

;
; Initial cache data for root domain servers.
;
. 99999999 IN NS NS.NIC.DDN.MIL.
 99999999 IN NS A.ISI.EDU.
 99999999 IN NS AOS.BRL.MIL.
 99999999 IN NS C.NYSER.NET.
 99999999 IN NS TERP.UMD.EDU.
 99999999 IN NS NS.NASA.GOV.
; Prep the cache (hotwire the addresses).
NS.NIC.DDN.MIL. 99999999 IN A 192.67.67.53
A.ISI.EDU. 99999999 IN A 26.3.0.103
A.ISI.EDU. 99999999 IN A 128.9.0.107
AOS.BRL.MIL. 99999999 IN A 192.5.25.82
C.NYSER.NET. 99999999 IN A 192.33.4.12
TERP.UMD.EDU. 99999999 IN A 128.8.10.90
NS.NASA.GOV. 99999999 IN A 192.52.195.10

```

## *named.local*

```

@ IN SOA ucbvax.Berkeley.Edu. kjd.ucbvax.Berkeley.Edu. (
 1 ; Serial
 10800 ; Refresh
 1800 ; Retry
 3600000 ; Expire
 86400) ; Minimum
 IN NS ucbvax.Berkeley.Edu.
1 IN PTR localhost.

```

*named.hosts*

```

;
; @(#)ucb-hosts 1.1 (berkeley) 86/02/05
;
@ IN SOA ucbvax.Berkeley.Edu. kjd.monet.Berkeley.Edu. {
 1.1 ; Serial
 3600 ; Refresh
 300 ; Retry
 3600000 ; Expire
 3600) ; Minimum

 IN NS ucbarpa.Berkeley.Edu.
 IN NS ucbvax.Berkeley.Edu.
localhost IN A 127.0.0.1
ucbarpa IN A 128.32.4
 IN A 10.0.0.78
 IN HINFO VAX-11/780 UNIX
arpa IN CNAME ucbarpa
ernie IN A 128.32.6
 IN HINFO VAX-11/780 UNIX
ucbernie IN CNAME ernie
monet IN A 128.32.7
 IN A 128.32.130.6
 IN HINFO VAX-11/750 UNIX
ucbmonet IN CNAME monet
ucbvax IN A 10.2.0.78
 IN A 128.32.10
 IN HINFO VAX-11/750 UNIX
 IN WKS 128.32.0.10 UDP syslog route timed domain
 IN WKS 128.32.0.10 TCP (echo telnet
discard sunrpc sftp uucp-path systat
daytime netstat qotd nntp link chargen
ftp auth time whois mtp pop rje finger
smtp supdup hostnames domain name server)

vax IN CNAME ucbvax
toybox IN A 128.32.131.119
 IN HINFO Pro350 RT11
toybox IN MX 0 monet.Berkeley.Edu
miriam IN MB vineyd.DEC.COM.
postmistress IN MR Miriam
Bind IN MINFO Bind-Request kjd.Berkeley.Edu.
 IN MG Ralph.Berkeley.Edu.
 IN MG Zhou.Berkeley.Edu.
 IN MG Painter.Berkeley.Edu.
 IN MG Riggle.Berkeley.Edu.
 IN MG Terry.pa.Xerox.Com.

```

## *named.rev*

---

```

;
; @(#)ucb-named.rev 1.1 (Berkeley) 86/02/05
;
@ IN SOA ucbvax.Berkeley.Edu. kjd.monet.Berkeley.Edu. {
 1.1 ; Serial
 10800 ; Refresh
 1800 ; Retry
 3600000 ; Expire
 86400) ; Minimum
 IN NS ucbarpa.Berkeley.Edu.
 IN NS ucbvax.Berkeley.Edu.
4.0 IN PTR ucbarpa.Berkeley.Edu.
6.0 IN PTR ernie.Berkeley.Edu.
7.0 IN PTR monet.Berkeley.Edu.
10.0 IN PTR ucbvax.Berkeley.Edu.
6.130 IN PTR monet.Berkeley.Edu.

```

## *Additional sample files*

---

The following sections contain an additional set of sample files for the name server. See the manual page entry for **named(ADMN)** for more information on specific entries in these files.

### *named.boot*

---

```

;
; Name Server boot file for Domain mynet.COM
;
; Type Domain Source file or Host
;
domain mynet.COM
primary mynet.COM /etc/named.data/mynet-hosts
cache . /etc/named.data/root.cache
primary 200.9.192.in-addr.arpa /etc/named.data/mynet-host.s.rev
primary 0.0.127.in-addr.arpa /etc/named.data/named.local

```

### *root.cache*

---

```
;
; Initial cached data for root domain servers.
;
; . 99999999 IN NS USC-ISIB.ARPA.
; 99999999 IN NS BRL-AOS.ARPA.
; 99999999 IN NS SRI-NIC.ARPA.
;
; Insert your own name servers here
;
; . 99999999 IN NS tandy.mynet.COM
;
; Prep the cache (hotwire the addresses)
;
tandy.mynet.COM. 99999999 IN A 192.9.200.2
;
; Root servers go here
;
tandy.mynet.COM. 99999999 IN A 192.9.200.2
;SRI-NIC.ARPA. 99999999 IN A 10.0.0.51
;USC-ISIB.ARPA. 99999999 IN A 10.3.0.52
;BRL-AOS.ARPA. 99999999 IN A 128.20.1.2
;BRL-AOS.ARPA. 99999999 IN A 192.5.22.82
```

### *named.local*

---

```
;
; Don't forget to increment the serial number in
; named.soa
;

$INCLUDE /usr/lib/named/mynet.soa
1 IN PTR localhost.
```

## *mynet-host.s.rev*

---

```

;
; Don't forget to increment the serial number in
; named.soa
;

$INCLUDE /usr/lib/named/mynet.soa

1 IN PTR merlin
2 IN PTR tandy
3 IN PTR tvi

```

## *mynet.soa*

---

```

;
; Don't forget to increment the serial number when you
; change this. SCCS or RCS might be a good idea here.
;

@ IN SOA tandy.mynet.COM. root.tandy.mynet.COM. (
 1.0 ; Serial
 3600 ; Refresh
 300 ; Retry
 3600000 ; Expire
 3600) ; Minimum
 IN NS tandy.mynet.COM.

```

## *Domain management*

---

This section contains information for starting, controlling, and debugging **named**(ADMN), the Internet domain name server.

### *Starting the name server*

---

The host name should be set to the full domain style name (that is, *monet.Berkeley.EDU*.) using **hostname**(TC). The name server is started automatically if the configuration file */etc/named.boot* is present. Do not attempt to run **named** from **inetd**(ADMN). This continuously restarts the name server and defeats the purpose of having a cache.



### *letchnamed.pid*

---

When **named** is successfully started, it writes its processID into the file */etc/named.pid*. This is useful to programs that want to send signals to **named**.

### */etc/hosts*

---

The **gethostbyname** library call can detect whether **named** is running. If it is determined that **named** is not running, it looks in */etc/hosts* to resolve an address. This option was added to allow **ifconfig(ADMN)** to configure the machine's local interfaces and to enable a system manager to access the network while the system is in single-user mode. It is advisable to put the local machine's interface addresses and a couple of machine names and addresses in */etc/hosts*, so the system manager can copy files from another machine with **rcp** when the system is in single-user mode. The format of */etc/hosts* has not changed. See *hosts(SFF)* for more information. Because the process of reading */etc/hosts* is slow, it is not advisable to use this option when the system is in multiuser mode. Note that all network software must be manually started in single-user mode.

### *Reload*

---

There are several signals that can be sent to the **named** process to have it do tasks without restarting the process. The **SIGHUP** signal causes **named** to read *named.boot* and reload the database. All previously cached data is lost. This is useful when you have made a change to a data file and you want **named**'s internal database to reflect the change.

### *Debugging*

---

When **named** is running incorrectly, look first in */usr/adm/syslog* and check for any messages logged by **syslog**. Next, send it a signal to see what is happening.

**SIGINT** dumps the current database and cache to */usr/tmp/named\_dump.db*. This should give you an indication as to whether the database was loaded correctly.

**NOTE** **named** is built with **DEBUG** defined. The following signals work when **DEBUG** is defined:

**SIGUSR1** turns on debugging. Each signal following **USR1** increments the debug level. The output goes to */usr/tmp/named.run*.

**SIGUSR2** turns off debugging completely

For more information on other signals and debugging BIND, refer to the named manual page.

## ***More BIND information***

---

The following Requests for Comments (RFC's) contain additional information about BIND:

| RFC    | Title and Information                                                                  |
|--------|----------------------------------------------------------------------------------------|
| RFC819 | The Domain Naming Convention for Internet User Applications, Su, Z., Postel, J., 8/82. |
| RFC882 | Domain Names - Concept and Facilities, Mockapetris, P., 11/83.                         |
| RFC883 | Domain Names - Implementation and Specification, Mockapetris, P., 11/83.               |
| RFC973 | Domain System Changes and Observations, Mockapetris, P., 2/86.                         |
| RFC974 | Mail Routing and The Domain System, Partridge, C., 2/86.                               |

These RFC's are available from:

Network Solutions  
 Attn: InterNIC Registration Services  
 505 Huntmar Park Drive  
 Herndon, VA 22070

## Chapter 5

# Gateways and routing

---

To exchange data between hosts that are attached to different networks, certain hosts, called “gateways”, are responsible for exchanging routing information and forwarding data from one network to another until the data reaches its final destination. Gateways can use several different protocols to exchange reachability and routing information among themselves. Some of these protocols are: EGP (Extended Gateway Protocol), BGP (Border Gateway Protocol), RIP (Routing Information Protocol), and the HELLO protocol.

In large interconnected networks that are connected to the Internet, each portion of the network that is under a separate local administration is called an Autonomous System (AS) and is assigned a unique number by InterNIC Information Services. Each AS is connected to a common core network via an exterior gateway. All the exterior gateways exchange host routing or reachability information among themselves using an exterior gateway protocol, such as EGP or BGP.

Within an AS there may be a single Local Area Network (LAN) or many interconnected networks that are linked via interior gateways. These gateways use an interior gateway protocol, such as RIP or HELLO, to exchange routing information.

This chapter describes two programs that are designed to exchange routing information and that you can run on a host to have it function as a gateway: the **gated**(ADMN) daemon and the **routed**(ADMN) daemon. The **gated** daemon allows a host to function as both an exterior and interior gateway, whereas the **routed** is intended only for interior gateways. Only one of these daemons can run on your host at any one time.

Table 5-1 summarizes the differences between the two:

**Table 5-1** routed/gated comparison

| Feature               | routed       | gated                  |
|-----------------------|--------------|------------------------|
| Protocol used         | RIP          | EGP, BGP, RIP, HELLO   |
| Autonomous system use | within an AS | within and between ASs |
| Route restricting     | none         | ASs can be excluded    |

## Running routed

If you wish to run **routed**, ensure that the file */etc/gated.conf* does not exist. If it does, you must remove it, move it to another directory, or rename it. Otherwise, the **gated** daemon will execute instead of **routed** the next time your host reboots. The **routed** daemon uses the Routing Information Protocol (RIP) to exchange routing information and the */etc/gateways* file to initialize static routes to distant networks. The */etc/gateways* file is described in the **routed** manual page.

When **routed** starts, it reads the */etc/gateways* file (if it exists), and installs the routes defined there into its routing table. It then broadcasts on each local network to find other hosts running **routed**. If such hosts exist, the routing daemons cooperate in maintaining a globally consistent view of routing within an Autonomous System. This view can be extended to include remote networks also running **routed** by setting up suitable entries in */etc/gateways*.

Another approach to routing to non-local networks is to define a default or wildcard route to a smart gateway and depend on the gateway to provide ICMP routing-redirect information to dynamically create a routing database. This is done by adding an entry to */etc/tcp* that invokes the **route(ADMN)** command, as in the following example:

```
/etc/route add default smart-gateway 1
```

The **route** command is used to manually manipulate routing tables. See the **route** manual page for more information.

The host uses the default route as a last resort in routing packets to their destinations. Assuming the gateway to which packets are directed can be reached to generate the proper ICMP routing-redirect messages, the host adds routing table entries based on the information supplied. This approach has certain advantages over using the **routed** daemon, but it is unsuitable in an environment where there are only bridges (for example, bridges do not generate ICMP routing-redirect messages.) Further, if the smart gateway goes down, there is no way to maintain service except to alter the routing table manually using the **route** command.

The host always listens to, and processes, ICMP routing-redirect messages, and so it is possible to combine both of the above facilities. For example, the **routed** daemon might be used to maintain up-to-date information about routes to local networks, while employing the wildcard routing techniques for distant networks.

## Running gated

Before you can run **gated** as your gateway daemon, you must first create the configuration file `/etc/gated.conf`. Before creating this file, you should be able to answer the following questions:

1. What is your Autonomous System ID (ASID)?
2. What protocol(s) will you use?
3. With which exterior and interior gateways will you be exchanging routing or reachability information?
4. What filtering controls are required for this gateway? For example, which AS's do you want to exclude?

Next, examine the information provided for each protocol that you wish to support from the appropriate protocol configuration files provided with your TCP/IP software distribution. The four protocols and related configuration files are:

| Protocol | Filename                      |
|----------|-------------------------------|
| EGP      | <code>/etc/gated.egp</code>   |
| BGP      | <code>/etc/gated.bgp</code>   |
| RIP      | <code>/etc/gated.rip</code>   |
| HELLO    | <code>/etc/gated.hello</code> |

Read the `gated.conf(SFF)` manual page carefully while examining these files. Using the provided files as a model, copy the appropriate statements into the `gated.conf` file and tailor them to your site requirements. Note that comments in the configuration file begin with a number sign “#” and statements are terminated with a semicolon “;”. Also, the definition, protocol, route, and control statements must be specified in the order just listed.

**NOTE** If the `/etc/gated.conf` file does not exist, **routed** is started as the gateway daemon instead of **gated** the next time your host reboots.

When first running **gated**, you should execute it with the `-n` option. This ensures that **gated** does not modify the kernel routing table. When you have finished testing the gateway and are satisfied that it is working correctly, restart the daemon without the `-n` option.

The basic flow of execution of **gated** when reading the configuration file is as follows:

1. Enable **traceoptions**.
2. If BGP or EGP is being used as the gateway protocol, establish the Autonomous System ID.
3. Enable or disable the appropriate protocols.
4. Specify the protocol attributes.

### *Sample configuration file*

---

Below is a sample *gated.conf* configuration file for a gateway connected to the Internet that has an IP address of 128.212.63.2 and that is part of an Autonomous System (AS) assigned an AS ID of 519. This gateway uses RIP to exchange routing information within the AS. This gateway also uses BGP to exchange routing information with another gateway on the Internet (the “peer gateway”) that has an IP address of 192.35.52.90 and that is part of an Autonomous System assigned an AS ID of 963.

---

```
traceoptions internal external route bgp ;
autonomoussystem 519 ;

rip yes {
 interface wdn0 ;
 trustedgateways 128.212.64.1 ;
} ;

bgp yes {
 peer 192.35.52.90 linktype horizontal asin 963 interface e3B0 ;
} ;

propagate proto bgp as 963 metric 10 {
 proto rip {
 announce all;
 };
};

propagate proto rip metric 2 {
 proto bgp as 519 {
 announce all;
 };
};
```

---

Following is a detailed explanation of all of the statements in the configuration file.

---

```
traceoptions internal external route bgp ;
```

---

This statement turns on tracing for internal errors, external errors and information messages. It also records changes to the **gated** routing table and all BGP packets that are sent and received.

---

```
autonomousssystem 519 ;
```

---

This statement indicates that the gateway is a part of the Autonomous System whose AS ID is 519.

---

```
rip yes {
 interface wdn0 ;
 trustedgateways 128.212.64.1 ;
} ;
```

---

This statement says that the Routing Information Protocol (RIP) is to be used to exchange routing information within our Autonomous System. The hardware interface to our Autonomous System is controlled by the SCO LLI driver *wdn0*. Finally, this statement says that we will update our RIP routing tables in response to new routing information only from the gateway with IP address 128.212.64.1.

---

```
bgp yes {
 peer 192.35.52.90 linktype horizontal asin 963 interface e3B0 ;
} ;
```

---

This statement says to use the Border Gateway Protocol (BGP) to exchange routing information with the peer gateway at IP address 192.35.52.90. The peer gateway is part of another Autonomous System (**linktype horizontal**) that has an AS ID of 963. Finally, our connection to the gateway is through the LLI driver *e3B0*.

---

```
propagate proto bgp as 963 metric 10 {
 proto rip {
 announce all;
 };
};
```

---

This statement says to advertise all of our RIP routes to the peer gateway using a metric of 10. This is necessary because BGP currently lacks a specification for advertising a default route to a peer gateway, so we must announce them all. Note that a metric must be specified to prevent the default metric from being used, because the default value indicates that the Autonomous System is unreachable.

**NOTE** The `gated` daemon uses the metric signifying “unreachable” as the default metric value for all supported protocols. Consequently, you should always specify a metric in the protocol statements that you place in the configuration file.

---

```
propagate proto rip metric 2 {
 proto bgp as 519 {
 announce all;
 };
};
```

---

This statement says to advertise all of our BGP routes to hosts within the Autonomous System using a metric of 2. Announcing all routes is useful in Autonomous Systems that have more than one external gateway.



Assuming the peer gateway uses the same interface cards and also uses RIP to advertise routes within its Autonomous System, its configuration file can be identical to the one just described, except for the following statements:

---

```

autonomoussystem 519 ;

bgp yes {
 peer 128.212.0.3 linktype horizontal asin 519 interface e3B0 ;
} ;

propagate proto rip metric 2 {
 proto bgp as 519 {
 announce all;
 };
};

```

---

Note that the Autonomous System ID and the IP address have been changed.

## More gated information

---

To get more information about gateway daemons, configuration files and commands, consult the following manual pages:

| Manual page     | Information provided              |
|-----------------|-----------------------------------|
| gated(ADMN)     | gated daemon                      |
| gated.conf(SFF) | gated configuration file          |
| routed(ADMN)    | routed daemon                     |
| route(ADMN)     | manually modifying routing tables |
| ripquery(ADMN)  | querying RIP gateways             |

To get more information about gateway protocols, consult the following Request for Comments (RFCs).

| Protocol | RFC        |
|----------|------------|
| EGP      | 911, 904   |
| BGP      | 1163, 1164 |
| RIP      | 1058       |
| HELLO    | 891        |

## 5. *Gateways and routing*

To obtain an Autonomous System ID for your network or collection of networks, or to obtain copies of RFC's, contact:

Network Solutions  
Attn: InterNIC Registration Services  
505 Huntmar Park Drive  
Herndon, VA 22070

## Chapter 6

# Configuring and using SNMP

---

The Simple Network Management Protocol (SNMP) monitors and controls TCP/IP-based networks. SNMP allows the retrieval and alteration of networking information maintained by hosts and routers attached to a network. A network administrator can use SNMP to diagnose and correct network problems from remote hosts.

By using the SCO SNMP implementation, a network administrator can gather information such as routing entries, interface status, and protocol statistics. If problems are encountered, the administrator can manipulate items such as the ARP cache and the routing table to add, delete, and modify incorrect entries.

This chapter describes SNMP commands and files, explains how to configure SNMP, and shows some examples of how you might

## Basic concepts

---

There are three basic components of SNMP: the protocol itself, described in detail in RFC 1157; SMI, the Structure of Management Information (RFC 1155); and MIB, the Management Information Base (RFC's 1156 and 1213). For information on obtaining these RFC's (Requests for Comments), see the section "More SNMP information" later in this chapter.

## The SNMP protocol

---

SNMP is a simple protocol with four basic operations: **get**, **get-next**, **set**, and **trap**. Each operation is encoded in a separate Protocol Data Unit (PDU). An additional PDU is defined for replying to **get**, **get next**, and **set** operations. The PDU's are carried through the network in the User Datagram Protocol (UDP) via UDP ports 161 and 162.

Although using an unreliable protocol such as UDP may seem non-intuitive at first, it has definite advantages. For example, sending UDP packets through a network that is damaging or corrupting packets is faster than establishing a TCP connection. This is because SNMP requires only one packet to get through for the request, and an additional one for the response. All TCP implementations would require three packets just to establish a connection.

### ***SMI: Structure of Management Information***

---

The Structure of Management Information (SMI), described in detail in RFC 1155, is a framework that describes the basic types of information that can be manipulated by SNMP. It provides a skeleton that specifies the basic format and hierarchy of management data, but does not describe the objects that can be managed. Instead, it describes the building blocks from which the managed objects are constructed.

A fundamental concept of the SNMP is the notion of *object identifiers*. An object identifier (OID) is a tag that allows a management entity to refer unequivocally to a particular object. Object identifiers are allocated in a tree fashion. The value of the object identifier is a series of integers that refer to a particular traversal of the object tree. Figure 6-1 shows the object identifier hierarchy.

The root of the OID tree has no label. Currently, there are three children of the root, CCITT(0), ISO(1), and Joint-ISO-CCITT(2). The ISO node has many children, one of which is org(3), which is allocated for international organizations. Under org(3) is the U.S. Department of Defense, dod(6), which has the child internet(1).

The name { *iso org dod internet* } is a symbolic representation for the integer series 1.3.6.1. Both refer to the object identifier of the Internet sub-tree. In practice, 1.3.6.1 can simply be referred to as "Internet". { *iso org dod internet* }, 1.3.6.1, and *internet* are all different ways of identifying the same object. In SNMP PDU's, only the numeric sequences are used.

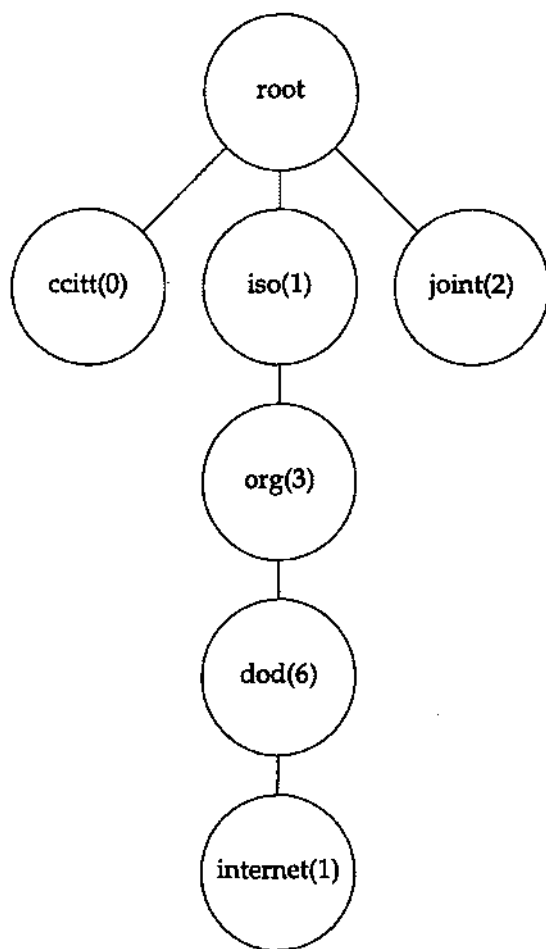


**Figure 6-1** Object Identifier hierarchy

To assure that object identifiers are unique, each organization is responsible for a particular section of the OID tree. Just as ISO and CCITT have responsibility for their portions, the Internet Activities Board (IAB) has responsibility for the Internet portion.

To allow vendors to support objects that may not be defined in the standard MIB, the IAB reserves a portion of the OID tree for **enterprises** MIB's. This provides vendors with the flexibility they need.

The SMI defines basic data types that make it convenient to describe managed objects. These types are: **ipAddress**, **Counter**, **Gauge**, **TimeTicks**, **NetworkAddress**, and **Opaque**. Readers who are interested in the precise semantics of these objects are encouraged to read the RFC for further information.

## ***MIB: the Management Information Base***

---

The Management Information Base (MIB), defined in RFC 1156 (MIB-I) and RFC 1213 (MIB-II), lists the standard objects that any SNMP implementation should support. MIB-II provides a superset of those objects found in MIB-I.

Each object in the MIB has two important characteristics. The first is its Object ID (OID). The second is its type. Object types are constructed from the fundamental types defined in the SMI.

The MIB is divided into logically related groups of objects. The important ones are: **system**, **interfaces**, **tcp**, **udp**, **ip**, **egp**, **icmp**, and **snmp**.

The **system** group contains general information about the network node. An example of this would be the physical location of the node, for example, "R&D Facility, 3rd floor machine room."

The **interfaces** group contains information about network interfaces, such as Ethernet® and point-to-point links. Information is kept about such items as interface status (for example, up or down), packet counts, and so on.

The rest of the groups contain information about the particular protocol to which they refer. This includes items such as the number of packets received with a particular protocol type, and number of packets received with incorrect checksums.

## ***SMUX (SNMP Multiplexing) protocol information***

---

On a typical system, an SNMP agent gathers the information pertinent to objects in the Internet standard MIB by reading variables from the kernel or stable storage (that is, files). Unfortunately, user-processes are often employed to perform many network services and the information the processes maintain cannot be easily accessed by the SNMP agent. SNMP cannot be used to manage the information maintained by user-processes unless the SNMP agent can communicate with the user-processes.

The SMUX protocol defines a mechanism by which multiple user-processes can communicate with the local SNMP agent. This enables the SNMP agent to manage objects whose data does not reside in areas directly accessible to it. This allows the set of objects managed by the local SNMP agent to vary dynamically.

For a user-process to be permitted to export a portion of the MIB (that is, establish a relationship with the local SNMP agent to manage objects for which the user-process maintains data), the user-process must be listed in the SNMP agent's `/etc/snmpd.peers` file; such a process is called an "SMUX peer." When a user-process wants to export a sub-tree of objects, it establishes a TCP

connection with the local SNMP agent, registers the sub-tree of objects it wants to export, and later on fields SNMP agent queries for objects in this sub-tree. The SNMP agent uses the TCP port 199 for listening to incoming requests from potential SMUX peers.

For more information on the SMUX protocol, refer to RFC 1227. Also, see the online reference manual pages `snmpd(ADMN)` and `snmpd.peers(SFF)`.

## Other concepts

---

In addition to the basic concepts described in the previous section, you should be familiar with the concepts described in the following sections.

### Agents and management stations

---

Systems using SNMP are divided into two categories: *management stations* (sometimes called clients) and *agents* (sometimes called servers). The management station is the system that issues a query; the agent is the system that is being queried.

Although there is no architectural reason that a system cannot have both types of functionality, the bulk of SNMP agents are usually found in dedicated routers and other specialized network equipment that is not capable of acting as a management station.

Using the SCO SNMP implementation, time-sharing systems and workstations can act as both agent and management stations.

### Traps

---

Although SNMP is normally used in a synchronous manner, in which systems poll each other periodically, it does include a mechanism for asynchronous events. In SNMP, these are referred to as “traps”. Traps are sent by agents to management stations when an abnormal event, such as a cold reboot, occurs.

### Authentication

---

In the current implementation of SNMP, authentication is based on a field in the PDU called the **community name**. When an agent receives a request, it validates the community name/IP address pair of the sender by comparing it against a specified list (found in `/etc/snmpd.comm`) of acceptable management stations. If the sender is not listed, or does not have the appropriate access permissions, the packet is discarded. Optionally, an **authentication trap** may be sent to a specified group of management stations, informing them that an invalid access has been attempted.

## Overview of the SCO implementation

---

As stated earlier, the SCO SNMP product provides the functionality of both agent and management station. A system running the software can monitor other systems, and be monitored as well.

**NOTE** The ability to operate as a true management station with full graphics capabilities is provided by third-party software that operates over SCO SNMP. Contact your software vendor for more information.

The SNMP Agent is implemented as a UNIX system daemon. There are three configuration files associated with the daemon. These are: */etc/snmpd.conf*, */etc/snmpd.comm*, and */etc/snmpd.trap*. The agent is started when the system enters the multiuser state (traditionally run-level 2). Upon startup, the agent reads configuration information from its configuration files, then begins listening for SNMP requests on the SNMP port. If configured to send traps, the agent also notifies the appropriate management stations that it is up. This is done by sending a “cold-start” trap to the systems listed in */etc/snmpd.trap*. Example 6-1 shows a sample of that file:

### Example 6-1 Sample snmpd.trap file

```
#
community address port
#
public 128.212.64.90 162
private 128.212.65.12 162
```

*/etc/snmpd.conf* contains the basic information that the agent needs as part of the **system** group. These are the types of SNMP and TCP/IP software, the object identifier of the agent, the name of the person responsible for the system, and the location of the system. Generally, only the contact field and the location field require changes. Example 6-2 shows a sample of */etc/snmpd.conf*:

### Example 6-2 Sample snmpd.conf file

```
#
descr=SNMPD Version 3.0 for SCO UNIX
objid=SCO.1.0
contact=Trevor Jones x 256
location=First Floor Computer Room
```

*/etc/snmpd.comm* contains a list of community/IP address pairs from whom the agent accepts queries. Along with each pair is an access field that controls whether access is allowed, and if so, whether the access granted is read-only or read-write. In addition, the special community *public*, with an address of *0.0.0.0*, restricts or allows access to all other members of the public community.



By default, */etc/snmpd.comm* contains one entry: `public 0.0.0.0 read`. This allows read access to all members of the public community.

Example 6-3 shows a sample of */etc/snmpd.comm*:

**Example 6-3 Sample snmpd.comm file**

```
#
community address access
#
public 0.0.0.0 none
private 128.212.64.90 read
ops 128.212.64.2 write
```

In addition to the agent, there are user commands that provide management station functionality to the system administrator. These are: `getid(ADMN)`, `getmany(ADMN)`, `getnext(ADMN)`, `getone(ADMN)`, `getroute(ADMN)`, `setany(ADMN)`, `snmpstat(ADMN)`, and `trap_rece(ADMN)`.

The `snmpstat(ADMN)` command provides the administrator with an easy way of retrieving information such as routing tables, address translation tables, interface status, and so on. Using `snmpstat`, a network administrator can quickly detect error conditions on a misbehaving network node.

The remaining commands provide the administrator with the ability to retrieve or modify arbitrary objects. Many vendors have their own enterprises MIBs in addition to the standard MIB. Because these continually evolve, it may well be the case that a particular node maintains information above that which is conveniently accessed via `snmpstat`. The additional commands can provide access to these enterprises MIBs.

## Configuring the SNMP agent

SNMP configuration consists of running a special configuration script, `mkdev snmp`. Your SCO TCP/IP Runtime System software, including the TCPRT and SNMPRT packages, must be fully installed and configured before you run this command. To configure SNMP:

1. Log in to the system as *root*.
2. Enter the following command at your prompt and press (Return):

```
mkdev snmp
```

**NOTE** If you did not install the complete TCPRT and SNMPRT packages, an error message appears and you return to your operating system prompt. Install the missing packages and run `mkdev snmp` again.

### 3. The following message appears:

Do you wish to install or delete the SNMP agent (i/d/q)

Type **i** to install the agent.

4. A screen appears describing the configuration process. After you read it, type **y** to continue the configuration process. Typing **n** or **q** exits the configuration script.
5. A screen appears describing two configurable system group entries found in */etc/snmpd.conf*, *sysContact* (the name of a person responsible for the system) and *sysLocation* (the physical location of the system). Accept the default values by pressing **<Return>**, or enter a value of your own and press **<Return>** when done.
6. A screen appears describing the Community Setup. Information you enter here is placed in the file */etc/snmpd.comm*, and it is read when your system receives requests from other computers.
  - Enter a community name and press **<Return>**.
  - Enter the IP address of a host within that community and press **<Return>**.
  - Enter the permissions for that host (**READ**, **WRITE**, or **ONE**), and press **<Return>**.
7. If you have more hosts to configure, repeat the previous step, entering the community name of the next host. After you have entered and configured all hosts, type **quit** at the community prompt.
8. A screen appears describing the Trap Systems Setup. Information you enter here is placed in the file */etc/snmpd.trap*, which contains a list of management stations to be notified when an abnormal event occurs on your system.
  - Enter a community name and press **<Return>**.
  - Enter the IP address of a host within that community and press **<Return>**.
9. If you have more hosts to configure, repeat the previous step, entering a community and its host IP address. After you have entered and configured all hosts, type **quit** at the community prompt.

The configuration is complete, and you return to your operating system prompt.
10. You can start the **snmpd** daemon now by entering **snmp start** at your prompt, or you can wait until the next time you bring the system into multiuser mode, when the daemon starts automatically.

The configuration script described above handles most configuration needs. However, you may want to edit the files by hand at a later time.

Full documentation on the SNMP agent and its configuration files can be found in the `snmpd(ADMN)`, `snmpd.comm(SFF)`, `snmpd.conf(SFF)`, and `snmpd.trap(SFF)` manual page entries.

## Using the SNMP commands

---

This section gives a brief overview of using some of the commands provided in the release. Before doing that, it is useful to explain some of the MIB variables used in the examples.

The following variables are from the **system** group: **sysDescr**, **sysContact**, and **sysName**. These are the description of the node, the name of the person responsible for the node, and the name of the node.

The following variables are from the **interfaces** group: **ifDescr**, **ifOperStatus**, **ifType**, and **ifPhysAddress** variables. These are the name, status, type, and physical address associated with a particular interface.

The following variables are from the **ip** group, and cover routing table information: **ipRouteDest**, **ipRouteNextHop**, and **ipRouteType**. These are the destination, next hop of the route entry, and the type of route (for example, direct or indirect).

The easiest command to use for retrieving specific variables is **getone(ADMN)**. The syntax of **getone** is **getone node community variable**. So, in order to ask node *grinch* for its system description, one would use the command **getone grinch public sysDescr.0**. This assumes that *grinch* accepts requests with a community of **public**. Example 6-4 shows two examples of the use of **getone**:

### Example 6-4 getone example

```
getone grinch public sysDescr.0

Name: sysDescr.0
Value: SNMPD Version 3.0 for SCO UNIX

getone grinch public sysName.0

Name: sysName.0
Value: grinch.i88.isc.com
```

The **getid(ADMN)** command retrieves a subset of the **system** group from a node. The syntax is **getid node community**. Example 6-5 shows an example of **getid**.

### Example 6-5 getid example

```
getid grinch public
```

```
Name: sysDescr.0
```

```
Value: SNMPD Version 3.0 for SCO UNIX
```

```
Name: sysObjectID.0
```

```
Value: SCO.1.0
```

```
Name: sysUpTime.0
```

```
Value: 178892
```

The **sysUpTime** variable is the number of clock ticks since the agent was restarted; each tick is 1/100th of a second.

The **getmany(ADMN)** command is useful for retrieving many variables at a time. The syntax of **getmany** is **getmany node community variable**. For example, the entire **system** group could be retrieved using the command **getmany node community system**. Or, information about all interfaces attached to the node could be retrieved with the command **getmany node community interfaces**. Example 6-6 shows an example of using **getmany** to get the **system** group.

**Example 6-6 getmany example**

```
getmany grinch public system

Name: sysDescr.0
Value: SNMPD Version 3.0 for SCO UNIX

Name: sysObjectID.0
Value: SCO.1.0

Name: sysUpTime.0
Value: 206730

Name: sysContact.0
Value: Trevor Jones x256

Name: sysName.0
Value: grinch.i88.isc.com

Name: sysLocation.0
Value: First Floor Computer Room

Name: sysServices.0
Value: 72
```

The **snmpstat**(ADMN) command provides a subset of the functionality of **getone**, and **getmany**, but has a much friendlier user interface. The **snmpstat** command can display routing, address translation, system, interface, and TCP connection information in a form that is easier to understand. The syntax of **snmpstat** is **snmpstat [options] [node] [community]**. If you specify a **community**, you must also specify a node name. If no **community** argument is specified, the default community of **public** is used. With no options specified, **snmpstat** displays the table of transport (TCP and UDP) endpoints. Table 6-1 lists the options available to **snmpstat**:

**Table 6-1 snmpstat options**

| Option | Description                                      |
|--------|--------------------------------------------------|
| -S     | show the <b>snmp</b> group                       |
| -a     | show the address translation table               |
| -i     | show statistics from the <b>interface</b> group  |
| -n     | print addresses numerically for active endpoints |
| -r     | show the routing table                           |
| -s     | show the <b>system</b> group                     |
| -t     | show all connections                             |

Example 6-7 shows an example of using **snmpstat** to retrieve the **system** group.

### Example 6-7 snmpstat system group example

```
snmpstat -s grinch public
System Group
Description: SNMPD Version 3.0 for SCO UNIX
ObjectID: SCO.1.0
UpTime: 1 hour, 1 minute, 40 seconds, 95 hundredths
Contact: Trevor Jones x256
Name: grinch.i88.isc.com
Location: First Floor Computer Room
Services: applications, end-to-end
```

**snmpstat** takes care of formatting the information in a way that is easier to understand. For example, **snmpstat** converts the raw timeticks of the agent uptime into a human-readable time.

Example 6-8 shows an example of using **snmpstat** to retrieve routing information.

### Example 6-8 snmpstat routing example

```
snmpstat -r grinch public
Routing table
Destination Gateway Metric Type Proto Interface
0.0.0.0 houdini 3 rem rip ni0
localhost localhost 0 dir local lo0
isc-i88-backbone grinch 0 dir local ni0
isc-i88-dev-net mcfeely 2 rem rip ni0
```

In this example, four routes are returned from *grinch*. Two of them are directly attached. These are the routes for *localhost* and the *isc-i88-backbone* subnet. Two others have been acquired by the RIP routing protocol from the hosts *houdini* and *mcfeely*.

The **setany(ADMN)** command can modify variables. Its basic syntax is **setany node community variable type value**. The legal values for *type* are given in Table 6-2. For a complete explanation of these types, refer to RFC 1155.

**Table 6-2 setany variable types**

| Object type       | setany argument |
|-------------------|-----------------|
| Integer           | -i              |
| Octet string      | -o              |
| Object identifier | -d              |
| IP address        | -a              |
| Counter           | -c              |
| Gauge             | -g              |
| Time_ticks        | -t              |
| Display string    | -s              |
| Null              | -n              |

Example 6-9 shows an example of using **setany** to alter the status of an interface. The status is verified before and after the change to ensure that the change took place correctly. Changing the status from 1 to 2 marks the interface down.

**Example 6-9 Altering interface status**

```
getone grinch public ifAdminStatus.2
Name: ifAdminStatus.2
Value: 1

setany grinch public ifAdminStatus.2 -i 2
Name: ifAdminStatus.2
Value: 2

getone grinch public ifAdminStatus.2
Name: ifAdminStatus.2
Value: 2
```

## Using SNMP to correct problems

---

In this section, several “real world” examples are discussed to demonstrate the capabilities of SNMP.

### Obtaining remote system contacts

---

Suppose that the administrator notices a number of local connection attempts to a restricted service (such as NNTP, the “USENET” News Transfer Protocol,) from an unknown host. The administrator can use `snmpstat` to determine some information about the host if the unknown host is running SNMP. Example 6-10 shows a possible scenario.

#### Example 6-10 Verifying the origin of a network connection

```
snmpstat grinch public
Active Internet connections
Proto Local Address Foreign Address (state)
tcp grinch.nntp bach.CS.ESU.EDU.1021 ESTABLISHED

snmpstat -s bach.CS.ESU.EDU
System Group
Description: SNMPD VERSION 9.4.0.0 SUN 3/260 - SUNOS3.5
ObjectID: SNMP_Research_UNIX_agent.9.4.0.3
UpTime: 4 days, 15 hours, 56 minutes, 39 seconds, 74 hundredths
Contact: Operations 1-999-555-1040
Name: bach.CS.ESU.EDU
Location: Comp Center
Services: applications, end-to-end
```

### Removing an incorrect routing entry

---

Suppose that due to a configuration error, a system in another part of the building had an incorrect entry in its routing table, and that this was causing problems communicating with other hosts in other company offices. The network administrator could use SNMP commands to correct the incorrect routing entry remotely. Example 6-11 shows a possible scenario. In the text, the command line is split with a backslash for clarity. It should be typed in as one line to the shell.



In addition to the new next hop, the administrator specifies the type of route. In this case, the route is an indirect or remote route (type 4), which was reported by `snmpstat` as `rem`.

#### Example 6-11 Correcting routing entries

```
snmpstat -r grinch public | grep 123.0.0.0
123.0.0.0 fonebone 2 rem rip ni0
```

```
setany grinch public ipRouteNextHop.123.0.0.0 \
-a 128.212.64.1 ipRouteType.123.0.0.0 -i 4
```

```
Name: ipRouteNextHop.123.0.0.0
Value: 128.212.64.1
```

```
Name: ipRouteType.123.0.0.0
Value: 4
```

The administrator could also have deleted the route by setting its `ipRouteType` to invalid (type 2). This is useful if a routing protocol reacquires the route information automatically.

## Marking an interface down

Suppose that a PPP interface attached to a router is reporting many errors. The administrator might wish to mark the link down while the phone company checked the line. SNMP could be used to do this remotely.

First, the error count can be obtained with `snmpstat`.

```
snmpstat -i grinch public
Interface statistics
```

| Name | Address      | Type     | InOctet  | InPkts | InErrs | IfMtu |
|------|--------------|----------|----------|--------|--------|-------|
| ni0  | grinch       | enetv2   | 18952186 | 18239  | 7036   | 1500  |
|      | 02000000ec6c | 10000000 | 4392148  | 4183   | 147    | 0     |
| lo0  | localhost    | loop     | 5832     | 64     | 0      | 2048  |
|      |              | 0        | 5832     | 64     | 0      | 0     |
| ppp0 | grinch       | ppp      | 3682304  | 14384  | 4800   | 296   |
|      |              | 9600     | 2122240  | 8290   | 2138   | 0     |

Next, the interface can be marked down using `setany`.

```
setany grinch public ifAdminStatus.3 -i 2
Name: ifAdminStatus.3
Value: 2
```

When the line is repaired, the interface can be reactivated in a similar fashion.

## Removing an incorrect ARP entry

---

Suppose that a misconfigured system had an incorrect IP address, causing bad ARP information to be installed on another node. SNMP could remove the offending address remotely. This can be very useful if the node in question has a very long timeout on the ARP cache, or some important connection is being established with the wrong machine. Example 6-12 shows a possible scenario.

### Example 6-12 Removing ARP entry

```
snmpstat -a grinch public
Address Translation table
Host Physical Address Flags Interface
128.212.64.9 8:0:14:40:1:21 dyna ni0
128.212.64.79 0:0:c0:6d:82:23 dyna ni0

setany grinch public ipNetToMediaType.1.128.212.64.9 -1 2

snmpstat -a grinch public
Address Translation table
Host Physical Address Flags Interface
128.212.64.79 0:0:c0:6d:82:23 dyna ni0
```

## More SNMP information

---

The following RFC's (Requests for Comments) contain more information about SNMP and other related protocols. To obtain these RFC's, contact Jon Postel, RFC Editor, USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA, 90292-6695.

RFC 768: User Datagram Protocol; J. Postel; 8/80

RFC 1058: Routing Information Protocol; Hedrick, C.L.; 6/88

RFC 1155: Structure and Identification of Management Information for TCP/IP-Based Internets.; Rose, M.T., McCloaghrie, K.; 5/90

RFC 1156: Management Information Base for Network Management of TCP/IP-Based Internets; McCloaghrie, K., Rose, M.T.; 5/90

RFC 1157: Simple Network Management Protocol (SNMP); Case, J.D., Fedor, M., Schoffstall, M.L., Davin, C.; 5/90

RFC 1213: Management Information Base for Network Management of TCP/IP-based Internets:MIB-II; McCloaghrie, K.; Rose, M.T.,eds.; 3/91

## Chapter 7

# Remote line printing

---

Remote Line Printing (RLP) allows UNIX system machines on a network to send print jobs to other UNIX system machines that have attached printers. The machines that send print jobs are called clients. The machines that directly control the printers are called print servers. Clients and print servers can be either an SCO system or a 4.3BSD system (or one of its derivatives, such as SunOS™).

The only restrictions on RLP are:

- printer classes are not supported
- not all line printer command options are supported on the client

RLP allows print jobs to be submitted from a client and printed by a print server, but it does not support remote administration of printers. To administer a printer attached to a print server, do one of the following:

- Log in to the print server at the console.
- Log in to the print server from another machine using terminal emulation.
- Issue printer administration commands from another machine using the `rcmd(TC)` command.

## Installing and removing RLP

---

Before installing RLP on your SCO system, be sure first to install TCP/IP. To install RLP, log in as *root*, then enter the command `mkdev rlp`. You now see the following display:

```
Remote printing configuration
```

```
Do you want to install or delete remote printing (i/d/q)?
```

You have three options:

- To quit and return to the shell prompt, enter **q**.
- To remove RLP after it has been installed, enter **d**. Once RLP has been removed, your local printing environment is restored and you are returned to the shell prompt.
- To install or reinstall RLP, enter **i**. Installation details are discussed in the rest of this section.

If you choose to install RLP, you now see the following messages:

```
Installing Remote Line Printing
```

```
Creating directories /usr/lpd/remote and /usr/spool/lpd and file /etc/printcap
Saving lp, cancel and lpstat commands to /usr/lpd/remote
Installing remote lp, cancel and lpstat commands in /usr/bin
Do you want to change the remote printer description file /etc/printcap (y/n)?
```

To define printers available through RLP, enter **y**. The `rlpconf` command is then executed. See “Setting up a client” and “Setting up a print server” later in this chapter for information on `rlpconf`.

To wait until later to define printers, enter **n**. You then see the message:

```
Run /etc/rlpconf to change the remote printer description file
```

Remember to run `rlpconf` at some point before using RLP. Otherwise, you will have no entries in the `/etc/printcap` file.

If the directories that RLP needs to do the installation already exist, you are asked the following:

```
Remote line printing working directory already exists.
```

```
Do you wish to continue installing the remote line printing system (y/n)?
```

Enter **y** to continue the installation. Enter **n** to stop and return to the shell prompt.

Next you see the following message:

```
Setting up rc scripts
```

This message tells you that the appropriate scripts are being defined to start up RLP every time the system is booted.

You are now asked the following question:

```
Do you want to start remote daemon now (y/n)?
```

To start the **lpd(ADMN)** daemon, enter **y**. This daemon must be running for RLP to work. To wait until later to start the daemon, enter **n**. See the manual page on **lpd** for more information.

## How RLP works

RLP operates somewhat differently from normal printing. First, a new file, called */etc/printcap(SFF)*, contains information about local and remote printers and related files needed by the print commands. Second, the usual SCO print commands (**lp**, **cancel**, and **lpstat**) are moved to the directory */usr/lpd/remote* for safekeeping, and new versions of these command are moved to */usr/bin*. These new commands consult the file */etc/printcap* to determine whether the print job is going to a local printer or a remote printer attached to a print server.

If the printer is local, printing occurs just as it would with the standard command. All standard command options are supported. This is because the new command simply invokes the standard command, now located in */etc/lpd/remote*. If the printer is remote, the command spools the print job and invokes the **lpd** daemon. This daemon packages the print job in 4.3BSD format and sends it over the network to the host specified in the */etc/printcap* file.

## Using RLP

---

To use RLP, you invoke one of the supported commands. There are actually two sets of commands: one for SCO clients and one for 4.3BSD clients and derivatives.

### SCO clients

---

On SCO clients, the RLP commands are **lp**, **cancel**, and **rlpstat**. These commands work just like their non-RLP counterparts when invoked with no options and when invoked with the following supported options:

```
lp -dprinter_name
lp -nnumber_of_copies
lp -ttitle_to_print_on_banner

rlpstat -aremote_printer_name (acceptance status)
rlpstat -oremote_printer_name (output requests status)
rlpstat -premove_printer_name (printer status)
rlpstat -vremote_printer_name (pathnames of associated devices)
```

Only the options listed above are supported for the **lp** and **rlpstat** commands when used for remote printing. If one or more unsupported options are specified, they are ignored and the print job is submitted without them. If printing is to be done locally, all of the options that are normally available for local printing are supported.

**NOTE** On an SCO client, you must install RLP before you can do remote printing. See “Installing and removing RLP” earlier in this chapter for more information.

See the **cancel(C)**, **lp(C)**, and **lpstat(C)** manual pages for more information on the supported options.

## 4.3BSD clients

---

On 4.3BSD clients and derivatives, use the standard printing commands **lpr**, **lprm**, and **lpq** to do remote printing. The following options are supported when sending a print job to a remote printer attached to an SCO system on which RLP has been installed:

**lpr -Pprinter\_name [-#number\_of\_copies] [-Ttitle] [-s]**

(The **-s** option uses symbolic links instead of copying the file to the spool directory.)

**lpq -Pprinter\_name**

**lprm -Pprinter\_name**

**lprm -Pprinter\_name job\_number**

Only the options listed previously are supported for the **lpr**, **lpq**, and **lprm** commands when used for remote printing. If one or more unsupported options are specified, they are ignored and the print job is submitted without them.

**NOTE** If the remote printer is attached to a print server running 4.3BSD or a derivative, the above restrictions do not apply. In this case, both the client and the print server are running 4.3BSD or a derivative. Remember that RLP is installed only on SCO systems.

If printing is to be done locally, all of the options that are normally available for local printing are supported.

## Setting up a client

---

The printer description file */etc/printcap* on the client machine provides routing and other information about printers.

**NOTE** If the printer to be set up is a local printer, that is, attached to the client for its private use, you should set it up following the instructions in "Setting up a print server" later in this chapter, but ignore instructions regarding the */etc/hosts.equiv* and */etc/hosts.lpd* files.

For a full description of the */etc/printcap* file format, see the **printcap(SFF)** manual page.

Here is an example of an entry in the */etc/printcap* file, followed by an explanation of each field. Note that fields are separated by colons. Also, this entry is actually one logical line continued over two lines on the screen with the continuation character "\ ".

```
sunlaser:\
:lp=:rm=sunburst:rp=sunlaser:sd=/usr/spool/lpd/sunlaser
```

Each of the fields is described below:

- The first field in this example is *sunlaser*, the name to be used on the client when referring to this printer. If the printer is remote, the name must be the same one used by the print server.
- The second field (before the first colon on the second line) is empty by default. When used, it defines the name of the error log file.
- The third field, starting with *lp=*, must be empty (that is, followed immediately by a colon) for remote printing. This field specifies the device name for a local printer.
- The fourth field, starting with *rm=*, specifies the network name of the print server. In this example, the print server is called *sunburst*.
- The fifth field, starting with *rp=*, specifies the name that the print server uses for the printer.
- The last field, starting with *sd=*, is the name of the spooling directory on the client. It is in this directory that print jobs are stored before being sent over the network to the print server. This directory is always located in */usr/spool/lpd* and bears the name of the printer. In this example, the spool directory is called */usr/spool/lpd/sunlaser*.



To configure a remote printer, execute the **rlpconf** command. This command edits the */etc/printcap* file for you. The following example illustrates the use of **rlpconf** to set up the *sunlaser* printer to produce an entry in the */etc/printcap* file like the one shown at the beginning of this section. Assuming you are logged in as *root*, invoke the **rlpconf** command. You see output similar to the following:

```

Remote Printing Configuration

Enter information for remote printers or local printers accepting
remote print jobs.

Please enter the printer name (q to quit): sunlaser

Is sunlaser a remote printer or a local printer (r/l)? r

Please enter the name of the remote host that sunlaser is
attached to: sunburst

Printer sunlaser is attached to host sunburst

Is this correct? (y/n) y
Would you like this to be the system default printer? (y/n) y

Make sure that your host name appears in sunburst's /etc/hosts equiv
or /etc/hosts.lpd file.
Make sure that sunlaser appears in sunburst's /etc/printcap file (BSD format)
Make sure that sunlaser has a spool directory on sunburst
Putting sunlaser in printer description file and creating spool dir
Updating LP information...done
Updating /usr/spool/lp/default...done

```

The above example shows how to configure your client to use a remote printer called *sunlaser* attached to the print server *sunburst*. After entering the name of the printer and the print server, you must verify the information just entered. If it is correct, enter **y**. If it is not, enter **n**, at which point you are asked if you want to try again. Answer **y** to be prompted again for the name of the printer and the print server. Answer **n** to exit and return to the shell prompt.

Continuing with the above example, you are asked if you wish to make the printer the system default. Answer **y** or **n**. If you answer **n**, users on the client that wish to route a print job to *sunlaser* must do so explicitly by using the **-d** option in the **lp** command.

Following this query, you are advised to verify certain things on the print server. The **rlpconf** command now checks that the printer is not already listed in */etc/printcap*. If it is, a message to that effect is displayed. If it is not, the printer information is added to */etc/printcap*.

In addition, the spool directory for the printer, */usr/spool/lpd/sunlaser* in this case, is created and configuration information indicating the name of the printer and the print server is stored in a file called *configuration*. This file is located in a subdirectory bearing the printer's name under */usr/spool/lp/admins/lp/printers*. In this example, the pathname of the file is */usr/spool/lp/admins/lp/printers/sunlaser/configuration*. Finally, if the printer is to be the system default, the file */usr/spool/lp/default* is updated to contain the name of the printer.

Once you have configured the printer, the screen clears and you see output similar to the following:

```
Remote Printing Configuration

Enter information for remote printers or local printers accepting
print jobs.

Please enter the printer name (q to quit): q
#
```

Enter **q** to exit **rlpconf** and return to the shell prompt.

Once you have finished configuring printers, ensure that the RLP daemon **lpd** is running. You can do this by entering the command **ps -fe** and verifying that the daemon appears in the listing.

## Setting up a print server

---

To set up a print server to process RLP requests, do the following:

- Ensure that the RLP daemon **lpd** is running. You can do this by entering the command **ps -fe** and verifying that the daemon appears in the listing.
- Ensure that all printers attached to the print server are working properly as local printers. See the documentation that accompanies the operating system running on the print server for information on how to set up local printer services.
- List all of the clients that are to have access to the print server in either the */etc/hosts.equiv* or the */etc/hosts.lpd* file using your favorite text editor. Listing a client in the latter file is more secure, in that only remote printer access is granted to the specified client. See the manual page that describes */etc/hosts.equiv* for information on the kind of access made available to hosts listed in that file.
- Execute **rlpconf** to create the */etc/printcap* file and the spool directories for printers attached to the print server.

Once you invoke **rlpconf**, you see output similar to the following:

```

Remote Printing Configuration

Enter information for remote printers or local printers accepting
print jobs.

Please enter the printer name (q to quit): sunlaser

Is sunburst a remote printer or a local printer (r/l)? l

Please enter the name of the device for sunlaser: /dev/laser

Printer sunlaser uses device /dev/laser

Is this correct? (y/n) y

Putting sunlaser in printer description file and creating spool
directory...done

```

First enter the name of a printer attached to the print server. Next, enter **l** in response to the second question to indicate that you are on a print server. Then, enter the full pathname for the device for the printer. You are now given the opportunity to verify the device filename for the printer. If it is correct, enter **y**. If it is not, enter **n**, at which point you are asked if you want to try again. Answer **y** to start from the beginning or answer **n** to exit and return to the shell prompt.

Once you indicate that the device filename is correct, the **rlpconf** command checks that the printer is not already listed in */etc/printcap*. If it is, a message to that effect is displayed. Otherwise, the printer information is added to the */etc/printcap* file and the spool directory for the printer, */usr/spool/lpd/sunlaser*, is created.

Once processing is complete for one printer, you are prompted to set up another printer. When you are done, enter **q** to exit **rlpconf** and return to the shell prompt.

## Deleting printcap entries

To delete an entry in the */etc/printcap* file, you must use your favorite text editor. Be sure to delete the entire entry, which is continued over two lines. Also, you should delete the spool directory */usr/spool/lpd/printer\_name* and the file */usr/spool/lp/admins/lp/printers/printer\_name/configuration* associated with the deleted printer. Finally, if the deleted printer is the system default printer, you should modify the file */usr/spool/lp/default* appropriately. See “Setting up a client” earlier in this chapter for more information.

## Chapter 8

# Synchronizing clocks

---

Making sure that all of the clocks running on the various hosts in a network or group of interconnected networks keep accurate time is a very important task. One reason is that many programs, including database management systems and configuration management systems, use timestamps as part of their processing. This task, often referred to as time synchronization, is much more difficult to achieve in the distributed processing environments often found in networks. Both the Time Synchronization Protocol (TSP) and the Network Time Protocol (NTP) can be used to achieve time synchronization.

Only one of these protocols should be used on a given network or group of interconnected networks. Using NTP is preferred because it keeps clocks synchronized with Coordinated Universal Time (UTC), the international time standard. To use NTP on your network, you must have access to the Internet. If you do not, your only alternative is to use TSP. Unlike NTP, TSP does nothing more than attempt to achieve local synchronization of clocks. In other words, it seeks to have all clocks on the locally interconnected networks keep the same time, but it does not attempt to keep clocks synchronized to an external, standard time reference. Each protocol is discussed in this chapter in a separate section.

## Time synchronization protocol

---

In this implementation of the Time Synchronization Protocol, the `timed(ADMN)` time daemon runs on each host on your network. These time daemons communicate with each other using a master-slave scheme in which one of the daemons is the master and all of the remaining daemons are slaves.

## 8. Synchronizing clocks

Each time daemon has two functions:

- it supports the synchronization of the clocks on the various hosts in the network
- it starts or takes part in the election of a new master if the current master becomes unavailable or stops functioning

The synchronization mechanism and the election procedure are described on the **timed** manual page. The rest of this section describes the administrative and technical issues of running **timed** at a particular site.

### *How the time daemon works*

---

A master time daemon measures the time differences between the clock of the host on which it is running and those of all the other hosts on the network. The master computes the network time as the average of the times provided by slaves on hosts whose clocks are not faulty. In this scheme, a host's clock is considered faulty if its time differs from that of the majority of the clocks on the network by more than a specified amount. The master time daemon then sends to each slave time daemon the correction that should be applied to its host's clock. This process is repeated periodically.

Because the correction is expressed as a time difference rather than an absolute time, transmission delays do not interfere with the accuracy of the synchronization. Whenever a host comes up on the network, it starts a slave time daemon that asks the master for the correct time and resets the host's clock before any user activity can begin. The time daemons are thus able to maintain a single network time in spite of the drift of clocks away from each other. The present implementation of **timed** is capable of keeping processor clocks synchronized to within 20 milliseconds, but some clocks, because of their design, cannot be adjusted to within an interval smaller than 1 second.

To ensure that the time synchronization service provided is continuous and reliable in the face of perturbing events, an election algorithm is available to elect a new master. Perturbing events include:

- the host running the current master time daemon crashes
- the master time daemon terminates because of a runtime error
- the network on which the master time daemon is running is partitioned or restructured

This algorithm allows slaves to detect that a master has stopped functioning and to elect a new master from among the slaves. Because the failure of a master time daemon results in only a gradual divergence of clock times among the slaves, the election of a new master need not occur immediately.

Running **timed** on gateways that connect distinct Local Area Networks (LANs) requires particular care because the time daemon may act as a “submaster”. A submaster is a time daemon that functions as a slave on one of the networks the gateway is connected to, and as a master on one or more of the remaining networks. Submasters are necessary to overcome the restriction of current transmission protocols preventing broadcasts from being transmitted over multiple physical networks.

A submaster classifies the networks that the gateway is connected to into the following three types:

- a slave network, which is a network on which the submaster acts as a slave. Only one of the networks can be a slave network.
- master networks, which are networks on which the submaster acts as a master
- ignored networks, which are networks that are neither slave networks nor master networks

The submaster tries periodically to become master on ignored networks, but gives up immediately once it detects that a master already exists on that network.

## Guidelines

While the synchronization algorithm is quite general, the election algorithm, which requires a broadcast mechanism, puts constraints on the kind of network on which **timed** can run. One restriction is that you should run **timed** only on networks with broadcast capability, possibly augmented with point-to-point links. Another restriction is that the maximum number of hosts that can be directly controlled by one master time daemon is 99.

If submasters are excluded, there is normally only one master time daemon running on a given LAN. During an election, only one of the slave time daemons becomes the new master. Not all hosts are suitable as masters, however. Because a master time daemon requires CPU resources proportional to the number of slaves it controls, you should consider running a master only on hosts with more powerful processors or lighter processing loads. Also, hosts with inaccurate clocks should not be used as masters. Note that this is a purely administrative decision. You are free to allow all of the hosts on your network to be a master.

At startup time, a time daemon running on a host connected to multiple networks may be instructed to recognize specific networks with the **-n** option or to ignore specific networks with the **-i** option. Note that both options cannot

be specified together. In other words, you can recognize certain networks or ignore certain networks, but you cannot do both. Typically, you would instruct the time daemon to ignore all networks except those that are under local administrative control.

If you run **timed** on several different LANs connected over gateways, you should avoid the situation where two gateways are both connected to the same two networks, and where the time daemon on those gateways play opposite master/slave roles on those networks. For example, suppose hosts A and B are gateways that are both connected to networks X and Y. If **timed** is started on both A and B with the **-M** option, it is possible for the time daemon on gateway A to become a slave on network X and the master on network Y, while the time daemon on gateway B can become a slave on network Y and the master on network X. If this kind of loop exists, the master time daemons do not function properly to establish a single value for time on both of the networks. It also causes the submasters to use large amounts of system resources in the form of network bandwidth and CPU time. This kind of loop can also be generated with more than two gateways, in the case where several LANs are interconnected.

### Options

---

The options for the **timed** command are:

- n network** recognize the named network
- i network** ignore the named network
- t** place tracing information in */usr/adm/timed.log*
- M** allow this time daemon to become a master. A time daemon started without this option will always be a slave.

### Daily operation

---

As part of the day-to-day operation of your network, you can use the **timedc(ADMN)** command to control the operation of the time daemons. For example, you can

- measure the differences between host clocks
- find the location of the master time daemon
- cause election timers on several hosts to expire at the same time
- enable or disable tracing of messages received by **timed**

Also, you can set the network date using the **rdate(ADMN)** command. See the manual pages on **timed**, **timedc**, and **rdate** for more information.

# Network time protocol

---

This section describes the Network Time Protocol (NTP) and offers guidelines on how to configure NTP. It includes example configurations, and describes how to get information on how well NTP is working, how to test NTP, how to tune it, and how to troubleshoot error conditions.

## Important terms

---

A number of terms describe important entities or activities that are part of NTP. These are defined here.

**client mode** the mode in which a host polls a time server that it might synchronize with, but it will not respond to polls from that time server. When a host is operating in this mode, the time server it is polling is said to operate in "server mode".

**Coordinated Universal Time**

the international standard reference time. It also corresponds to the local time at zero longitude. The standards pertaining to the definition and maintenance of Coordinated Universal Time, which effectively replaced Greenwich Mean Time (GMT) on 1 January 1972, are promulgated in Recommendation 460 of the International Consultative Committee for Radio (Comite Consultatif International de Radiodiffusion or CCIR) of the International Telecommunications Union. The CCIR is located at 2, rue de Varembe, 1211 Geneva 20, Switzerland. Responsibility for time standards in the United States rests with the Time and Frequency Division of the National Institute of Standards and Technology (NIST) in Boulder, Colorado.

**dispersion** a measure, in seconds, of how scattered the time offsets have been from a given time server

**drift** a measure, in Hertz per second, of how quickly the skew of a clock is changing. See also "skew".

**host** any computer connected to the network

**Internet** the collection of interconnected networks that grew out of ARPANET, the Defense Advanced Research Projects Agency network, and that use TCP/IP to communicate to function as a single cooperative network

**poll** the sending of an NTP packet from a host to an NTP time server to request the current time. The server responds by recording the current time in the packet, then sending it back to the originating host. See also "NTP packet."



- NTP packet** a message sent over the network that conforms to the Network Time Protocol format. This format includes space for recording the current time. See also "poll".
- primary server**  
another name for a stratum 1 server. See also "stratum".
- roundtrip delay**  
the time it takes for a host to send an NTP packet to another host and get an NTP packet back from that host in reply
- secondary server**  
another name for a stratum 2 server
- server mode** the mode in which a time server allows itself to be polled by a host (the client) that wishes to synchronize with it. In this mode, if the time server polls the client to try to synchronize with it, the client does not respond. In this case the client is said to operate in "client mode".
- skew** a measure, in Hertz, of the difference between the actual frequency of a clock and what its frequency should be to keep perfect time. See also "drift".
- slew** to adjust gradually the time of a clock until it tells the correct time. Compare with "step".
- step** to change the time of a clock to the correct time with no intermediate adjustments. Compare with "slew".
- stratum** the distance a host running the `xntpd` time daemon is from an external source of Coordinated Universal Time (UTC). A stratum 1 server has direct access to an external source of UTC, such as a radio clock synchronized to a standard time signal broadcast. In general, a stratum  $n$  server is  $n-1$  network hops away from a stratum 1 server. For example, a stratum 4 server is 3 hops away from a stratum 1 server. Also, a stratum  $n$  server is at a higher stratum than a stratum  $n-1$  server. For example, a stratum 3 server is at a higher stratum than a stratum 2 server, and at a lower stratum than a stratum 4 server. See also "time daemon".
- symmetric active mode**  
the mode in which a host has configured itself to poll a time server that it might synchronize with. In this mode, the host also allows itself to be polled by that time server.
- symmetric passive mode**  
the mode in which a time server is polled by a host that has configured itself in "symmetric active mode". In this mode, the time server can also poll that host.

**synchronization subnet**

a collection of hosts that synchronize time with each other. The top layer of the subnet consists of stratum 1 servers.

**synchronize clocks**

to set two clocks to the same time and ensure that they are running at the same “speed”. The speed at which a clock runs is determined by its frequency, that is, how often it “ticks” to the next fraction of a second. The design of a particular clock determines how small that fraction is.

**synchronize with a host**

to synchronize the local clock with another host’s clock, either by stepping or slewing the local clock to the time reported in the NTP packet received from the most recent poll of that host.

**time client**

a host running the `xntpd` time daemon that requests a time server to furnish it with that server’s best estimate of Coordinated Universal Time. Hosts running at a stratum higher than 1, except for those at the highest stratum, typically function as both time clients (polling same-stratum or higher-stratum servers) and time servers (furnishing the current time to other hosts). Compare with “time server”.

**time daemon**

the program running on a host that synchronizes the host’s hardware clock to Coordinated Universal Time in accordance with the protocols known as the Network Time Protocol. The name of this program is `xntpd`.

**time server**

a host running the `xntpd` time daemon that, upon request, furnishes its best estimate of Coordinated Universal Time. Compare with “time client”.

**trap receiver**

a program that listens for NTP mode 6 packets sent from another host. The host that “springs the trap” by sending the mode 6 packet to the trap receiver does so in response to the occurrence of an exception. These exceptions include the following:

- host reboots
- host detects that its clock has been reset
- host detects that the stratum of some other host in the synchronization subnet has changed
- host detects that a new host has been added to the synchronization subnet
- host detects that some other host in the synchronization subnet has become unreachable

- host detects that some other host in the synchronization subnet has been authenticated
- host selects another host to synchronize with
- host detects that a system call has failed

### Overview

---

The Network Time Protocol (NTP) defines a set of procedures for synchronizing clocks on hosts connected to a network with access to the Internet. Some of the hosts act as time servers, that is, they provide what they believe is the correct time to other hosts. Other hosts act as clients, that is, they find out what time it is by querying a time server. Some hosts act as both clients and time servers, because these hosts are links in a chain over which the correct time is forwarded from one host to the next. As part of this chain, a host acts first as a client to get the correct time from another host that is a time server. It then turns around and functions as a time server when other hosts, acting as clients, send requests to it for the correct time.

The network time daemon `xntpd`(ADMN) is the program running on each of the hosts in the network that, following the conventions defined in the protocol, attempts to establish the correct time. It does this by using the best available source of time to synchronize the host clock with the time at zero longitude, known as Coordinated Universal Time or UTC.

### Guidelines

---

If your network is small, consisting of around a dozen hosts, you need only a single NTP time server on your network. Locate three time servers elsewhere on the Internet from which your time server can get the correct time. If possible, pick two stratum 1 servers and one stratum 2 server. Ideally, you should choose to get the time from servers that are closest to your host in terms of roundtrip delay. In other words, you should prefer servers for which the amount of time it takes to send a request for the current time and the time it takes to receive a response is smallest. However, there is no great harm if you select servers for which you do not know the roundtrip delay.

If, however, your network is large, consisting of dozens or hundreds of hosts, you should set up a larger number of time servers. A good rule of thumb is to have one time server for every dozen hosts. In this case, you first establish a few stratum 2 time servers, then you establish the remaining time servers as stratum 3 majority servers. Finally, all of the remaining hosts, which constitute the vast majority of the hosts on the network, become stratum 4 hosts.

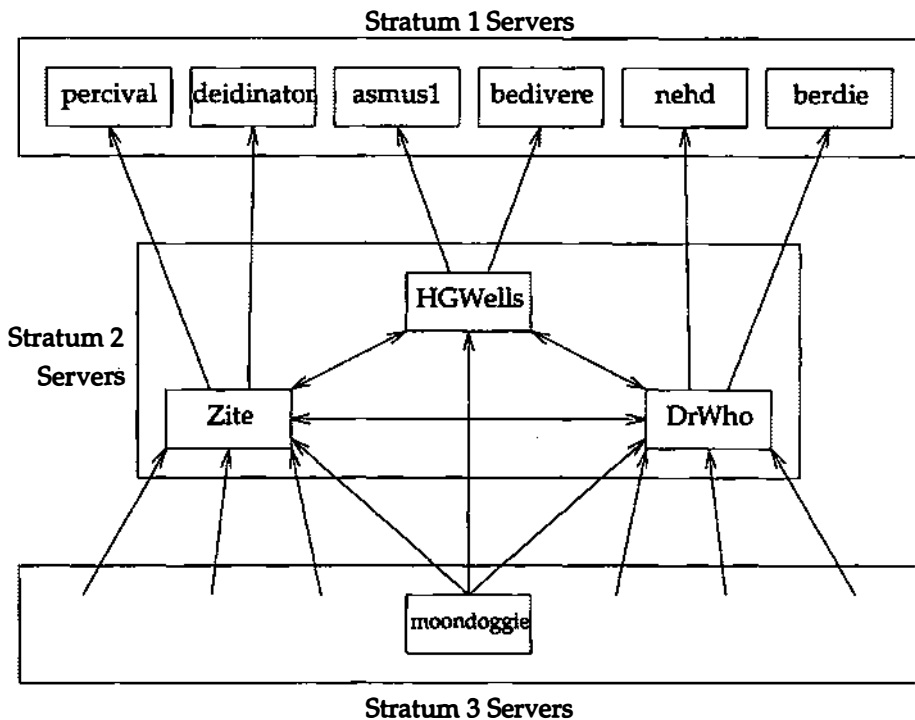
Below are guidelines for setting up NTP on a network of medium size.

- Choose three of your local hosts to operate as stratum 2 servers. Then choose six stratum 1 servers and configure each of your stratum 2 servers to poll a distinct pair of the stratum 1 servers. If possible, see to it that each stratum 2 server polls at least one stratum 1 server for which the roundtrip delay is small. You should also configure each of the stratum 2 servers to poll both of the other stratum 2 servers. Thus, each stratum 2 server is configured to poll two stratum 1 servers elsewhere on the Internet and two stratum 2 servers connected to the local network.
- From the remaining hosts, choose those you would like to operate at stratum 3. This might be all of the rest if you want to synchronize at most a couple of dozen hosts, or you might choose hosts such as file servers or those with good clocks if you need to synchronize many more hosts. Configure each stratum 3 server to poll the three stratum 2 servers.
- Configure all remaining hosts to either poll or listen for broadcast packets from three or four of the nearby stratum 3 servers.
- Avoid configuring hosts beyond stratum 4. Doing so makes the network unnecessarily complex and harder to manage.
- Do not configure a host to poll another host at the same stratum, unless the latter is directly polling lower stratum servers that the former does not.
- Do not configure hosts to poll other hosts at higher strata.

### *An example synchronization subnet*

Assume that hosts *Zite*, *HGWells*, and *DrWho* are stratum 2 servers in the imaginary *sci.com* network. Assume also the following:

- *Zite* polls two stratum 1 servers, *percival.cs.purdue.edu* and *deidinator.psy.chi.il.us*.
- *HGWells* polls two stratum 1 servers, *asmus1.genetics.uga.edu* and *bedivere.cs.purdue.edu*.
- *DrWho* polls two stratum 1 servers, *nehd.gov* and *berdie.athens.ga.us*.
- *Zite* polls *HGWells* and *DrWho*.
- *HGWells* polls *Zite* and *DrWho*.
- *DrWho* polls *Zite* and *HGWells*.
- *Moondoggie* and the rest of the machines at *sci.com* are at stratum 3 and listen for broadcast NTP packets from the three stratum 2 hosts.



A → B means A polls B or A listens for broadcast NTP packets from B.  
 C ↔ D means C polls D and vice versa.

### The NTP configuration file

The configuration file `/etc/ntp.conf` contains information that the `xntpd` daemon uses at startup, including:

- what are the possible hosts to synchronize with
- how to select which host to synchronize with
- what restrictions exist on other hosts communicating with the local host
- whether or not to broadcast NTP packets on the local network
- whether or not to listen for broadcast NTP packets
- what the roundtrip delay is between the local host and the host on this network (if any) that is broadcasting NTP packets
- where the *driftfile* is located
- whether or not to monitor NTP connections
- whether or not to allow runtime reconfiguration of the local host

The syntax of the configuration file is fairly free format. Comments begin with a “#” character and extend to the end of the line. Comments may be inserted freely and blank lines are ignored. Configuration statements include an initial keyword followed by arguments that are separated by white space (blanks or tabs). These statements may not be split over multiple lines. A list of the configuration statements with their arguments appears below. Optional arguments are delimited by “[” and “]”, and alternatives are separated by “|”.

## Configuration statements

Below is a list of the configuration statements you can use in the NTP configuration file and an explanation of each statement. The term “local host” in the explanations refers to the host that you are configuring with these statements.

**peer *host\_address* [ *key key\_ID* ] [ *version version\_number* ] [ *minpoll* ]**

specifies that the host with IP address *host\_address* is to be polled in symmetric active mode. The *key* option indicates that all packets sent to the host at *host\_address* are to include authentication fields encrypted using the specified key ID. The default is not to include authentication fields. The *version* option allows you to specify the version number to use for outgoing NTP packets. The allowed values for *version\_number* are 1 and 2. If this option is omitted, version 2 is assumed. The *minpoll* option specifies that the polling interval should be kept at the minimum value allowed. This maximizes the frequency with which the host is polled and should, therefore, be used only for debugging purposes.

**server *host\_address* [ *key key\_ID* ] [ *version version\_number* ] [ *minpoll* ]**

specifies that the host with IP address *host\_address* is to be polled in client mode. See the *peer* statement for an explanation of the remaining arguments.

**broadcast *IP\_address* [ *key key\_ID* ] [ *minpoll* ]**

specifies that *xntpd* broadcast NTP packets to the address *IP\_address*. This should be the broadcast address of your local network. See the *peer* statement for an explanation of the remaining arguments.

**precision *integer***

indicates the precision of the local clock as 2 raised to the *integer* power. The value of *integer* should be negative to indicate a fraction. See “Testing and tuning” later in this chapter for more information.

**driftfile *filename***

specifies the name of the file used to record the drift that *xntpd* has computed. Drift is also known as “frequency error”.

### **monitor *flag***

indicates whether or not to enable the `xntpd` monitoring function. The allowed values for *flag* are `yes` and `no`. A valid `requestkey` statement must be included in the configuration file for monitoring to function. The default value for *flag* is `no`.

### **broadcastclient *flag***

indicates whether or not the local host should listen for, and attempt to synchronize with, NTP packets broadcast by time servers on the local network. The allowed values for *flag* are `yes` and `no`. The default is `no`.

### **broadcastdelay *seconds***

specifies the roundtrip delay in seconds between the local host and the time server that is broadcasting NTP packets. Note that you specify a roundtrip delay, even though NTP cares only about one-way delay when it actually computes the correct time. The default value for *seconds* is 0.008.

### **keys *pathname***

specifies the pathname of the *keys* file. At installation the value of *pathname* is `/etc/ntp.keys`. This file must exist for the `requestkey` and `controlkey` configuration statements to be valid. See “The keys file” later in this chapter for more information.

### **requestkey *key\_ID* ...**

specifies the key IDs used to authenticate runtime reconfiguration requests made by `xntpd` and `xntpres` using mode 7 control messages. The *keys* file must exist and properly define all of the listed key IDs for this statement to be valid. To disable runtime configuration of `xntpd`, omit this statement from the configuration file. Doing so has the effect of disabling `xntpd` and `xntpres` on your local host.

### **controlkey *key\_ID* ...**

specifies key ID used to authenticate runtime reconfiguration requests made by `ntpq` and by trap receivers using mode 6 control messages. The *keys* file must exist and properly define all of the listed key IDs for this statement to be valid. To cause mode 6 control messages to be ignored, omit this statement from the configuration file. Doing so has the effect of disabling `ntpq` and trap receivers on your local host.

### **restrict *host\_address* *mask* *address\_mask* *flag* ...**

specifies various restrictions on how certain hosts and networks can interact with the local host. See “Address and mask facility” later in this chapter for more information.

**trap *host\_address* port *port\_number* interface *interface\_address***

configures a trap receiver at address *host\_address* and port *port\_number*. The value of *interface\_address* must be the IP address of the local host. The host at *host\_address* must be listening for NTP mode 6 control messages or it will not respond to the trap signal sent by the local host. This means that the host at *host\_address* must have the **controlkey** statement in its configuration file and both that host and the local host must share at least one of the keys listed in that **controlkey** configuration statement. See “The keys file” later in this chapter for more information.

**maxskew *seconds***

sets the system maximum skew parameter to the number of seconds specified. See “Testing and tuning” later in this chapter for more information.

**select *algorithm\_number***

specifies one of five weighted algorithms (numbered 1 through 5) used to select which time server the local host synchronizes with, out of the set of time servers from which the local host is receiving NTP packets. The default value for *algorithm\_number* is 1.

All of the algorithms attempt to select an accurate source time source (a “truechimer”) without excessive hopping from one time source to another. These algorithms take into account the following pieces of information:

- stratum of time source
- transmission delay between host and time source
- offset of clock on time source
- dispersion of clock on time source

In comparison to algorithm 1, algorithm 2 gives more weight to low-stratum time sources (algorithm 3 even more so), whereas algorithm 4 gives more weight to time sources for which the roundtrip delay is small (algorithm 5 even more so).

The basic algorithm is described in RFC 1119 Section 4 and updated in RFC 1129 Section 4.2. The weights associated with the different algorithms are: 0.75 for algorithm 1; 0.6875 for algorithm 4; and 0.625 for algorithm 5. Compared to algorithm 1, algorithm 2 reduces the weight of high stratum hosts by about 9%, while algorithm 3 reduces the weight of high stratum hosts by about 18%.

**resolver *pathname***

indicates the full pathname of the **xntpres** program, which resolves network names to IP addresses. When NTP is installed,



the pathname of the `xntpres` program is `/etc/xntpres`. If for some reason you move `xntpres` to another directory, use the pathname corresponding to its new location. You must also include the `requestkey` statement in the configuration file for `xntpres` to operate.

### Example `ntp.conf` file

The following is an example of a minimal `ntp.conf` file. The name of the host in this example is `zite.sci.com`.

---

```
#
Peer configuration for 128.212.66.90 (zite.sci.com)
(expected to operate at stratum 2)
#
peer 128.211.1.4 # percival.cs.purdue.edu (stratum 1 server)
peer 192.35.53.67 # deidinator.psync.chi.il.us (stratum 1 server)

peer 128.212.66.67 # HGWells.Sci.com
peer 128.212.66.118 # DrWho.Sci.com

driftfile /etc/ntp.drift
```

---

The four `peer` statements tell `xntpd` which servers to poll. The `driftfile` statement tells `xntpd` the name of the file in which to store the frequency error of the system clock. You are strongly encouraged to include this statement in the configuration file. This server is expected to operate as a stratum 2 server, because the first two servers have hardware clocks and are, therefore, stratum 1 servers.

### The `keys` file

The `keys` file contains a list of numeric key IDs and key values. These IDs and values are used to verify that mode 6 and mode 7 NTP packets should be processed. For example, when running the `xntpd` program, you must supply a valid key ID in response to the `Keyid` prompt and its associated key value in response to the `Password` prompt. See "Further examples" later in this chapter for sample displays of this.

In addition to a key ID and its associated value, each entry also contains a one-letter code indicating the type of the key value. The precise format of an entry in the key file is:

---

| <i>key_ID</i> | <i>key_type</i> | <i>key_value</i> |
|---------------|-----------------|------------------|
|---------------|-----------------|------------------|

---

The three fields shown above are separated by any combination of blanks and tabs. Comments may appear on any line and must begin with the number sign "#". The fields are:

- key ID, which is an arbitrary, unsigned 32-bit number, written in decimal. The range of possible values is zero through 4,294,967,295. Key IDs are specified by the `requestkey` and `controlkey` statements in the configuration file.
- key type, which identifies the type of `key_value`. There are three key types:
  - The first type is an ASCII character string between 1 and 8 characters long. The 7-bit ASCII representation of each character is used for each octet of the key. In this format, the high-order bit of each octet is the parity bit. The one-letter code for this type is "A".
  - The second type is a hexadecimal number written in the Department of Defense National Security Agency (NSA) Data Encryption Standard (DES) standard format. In this format, the low-order bit of each octet is the parity bit. The one-letter code for this type is "S".
  - The third type is a hex number written in NTP standard format. In this format, the high-order bit of each octet is the parity bit. The one-letter code for this type is "N".
- key value, which is a 56-bit DES encryption key that is written as 8 octets. The key value always uses odd parity.

Note that you will find it easier to specify ASCII key values, particularly for keys that are used to verify `xntpd` requests. Because this file contains authorization data, you are strongly urged to limit read permission for this file. In particular, you should remove read permission for `other`.

Below is a sample `keys` file:

---

```
#
Keys 4, 6 and 22 are actually identical at the bit level.
#
4 A DonTTelL
6 S 89dfdca8a8cbd998
22 N c4ef6e5454e5ec4c

#
The following three keys are also identical at the bit level.
#
100 A SeCReT
1000 N d3e54352e5548080
100000 S a7cb86a4cba80101
```

---

### *The clock.txt file*

The file */pub/ntp/doc/clock.txt* on host *louie.udel.edu* is your basic source of information on the location of stratum 1 and good quality stratum 2 servers. Among other things, this file also lists:

- sources for timecode receivers
- sources for precision oscillators and timing receivers
- call letters, location, frequencies, and coordinates for timecode transmitters
- an algorithm for great-circle distance and bearing computation
- information on timecode transmitters in Germany, the United Kingdom, and France

The information on primary and secondary servers in *clock.txt* has the following format:

---

**Hostname** (*IP address*)

**Location:** *organization, geographic location*

**Synchronization:** *machine type, clock type, synchronization protocol*

**Service area:** *networks served*

**Access policy:** *open or closed to the public*

**Contact:** *name and email address of contact person*

**Note:** *miscellaneous information*

---

One way to get the *clock.txt* file is to use *ftp*(TC). Below is a sample session showing how you might download this file to your host.

```
ftp louie.udel.edu
Connected to louie.udel.edu
220 louie.udel.edu.
Name (louie.udel.edu:muckel): anonymous
331 Guest login ok, send ident as password.
Password: guest
230 Guest login ok, access restrictions apply.
ftp> cd /pub/ntp/doc
250 CWD command successful.
ftp> get clock.txt
200 PORT command successful.
150 ASCII data connection for clock.txt (128.212.66.118,1198) (57002 bytes).
local: clock.txt remote: clock.txt
58409 bytes received in 8 seconds (7.1 Kbytes/s)
ftp> quit
221 Goodbye.
```

## The driftfile

The *driftfile* entry in */etc/ntp.conf* tells *xntpd* the name of the file where it can find and store the clock drift, also known as frequency error, of the system clock. If the file exists at startup, it is read and the value used to initialize *xntpd*'s internal value of the frequency error. The file is updated once every hour by *xntpd*. It usually takes a day or so after the daemon is started to compute a good estimate of the clock drift. Once the initial value is computed it will change only by relatively small amounts during the course of continued operation. Because *xntpd* needs a good estimate to synchronize closely to its server, there should always be a *driftfile* entry in */etc/ntp.conf*.

## Association modes

There are several different modes that hosts can be directly configured to operate in with respect to synchronization with other hosts: client mode, broadcastclient mode, and symmetric active mode.

In client mode, a host polls other hosts to get the current time. From among all of the hosts polled, the local host selects one with which to synchronize. To configure your host this way, include a **server** statement in your host's configuration file. The name or IP address of each time server to be polled must be specified in the **server** statement.

In broadcastclient mode, a host does no polling at all. Rather, it listens for NTP packets broadcast over the network. From among all of the broadcasting hosts, the local host selects one with which to synchronize. To configure your host this way, include a **broadcastclient yes** statement in your host's configuration file. For this mode to function, the local time servers must be configured in broadcast mode with the **broadcast** configuration statement.

Finally, in symmetric active mode, a host polls other hosts and responds to polls from those hosts. In addition, hosts retain time-related information about the hosts with which they are communicating. To configure your host this way, include a **peer** statement in your host's configuration file. The name or IP address of each time server must be specified in the **peer** statement.

A general guideline is to configure all hosts, except those at the highest strata of the synchronization subnet (those furthest away from stratum 1 servers), to operate in symmetric active mode. For those hosts that are at the highest strata, you have a choice: configure them to operate either in client mode or in broadcastclient mode. You should consider opting for broadcastclient mode if your hosts are on a high-speed LAN that supports broadcasts efficiently, especially if the hosts number more than twenty or so.

If you do opt for broadcastclient mode on those hosts, you must configure the time servers on the local network to be both in broadcast mode (to send broadcast NTP packets on the local network) and in symmetric active mode (to poll hosts at lower strata).

### *Address and mask facility*

The address and mask configuration facility adds various restrictions or erect barriers between your host and other time servers. A typical statement in the configuration file looks as follows:

**restrict IP\_address mask IP\_address\_mask flag1 flag2 ...**

Each statement adds an entry to an internal list maintained by **xntpd**. Each entry in this list contains the list entry address (the IP address following **restrict**), the address mask, and the flags. Below is a list of all of the flags and their meaning:

|                    |                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>ignore</b>      | indicates that all packets from hosts matching this entry will be ignored                                                     |
| <b>noquery</b>     | indicates that your host will not respond to mode 6 and 7 packets sent from hosts with a matching address                     |
| <b>nomodify</b>    | indicates that your host will not allow itself to be reconfigured by hosts with a matching address                            |
| <b>notrap</b>      | indicates that your host will not allow hosts with a matching address to register as a trap receiver                          |
| <b>lowpriotrap</b> | indicates that your host will give hosts with a matching address a low priority for the use of traps                          |
| <b>noserve</b>     | indicates that your host will not give the time to hosts with a matching address                                              |
| <b>nopeer</b>      | indicates that your host will not attempt to get time from any host with a matching address                                   |
| <b>notrust</b>     | indicates that hosts matching this entry, while treated normally in other respects, should not be trusted for synchronization |

When `xntpd` receives a packet, it compares the address of the host that sent the packet (the source address) with each entry in the internal list. Whenever the following relation (expressed in C language syntax) is true, a match occurs.

---

```
(source_address & address_mask) == (list_entry_address & address_mask)
```

---

In words, the source address and the address mask are logically ANDed together bitwise, the list entry address and the address mask are logically ANDed together bitwise, and the two results compared for equality. If the results are equal, a match has occurred. To establish default restrictions that apply to all hosts for which no match is found, include a statement like the following in the configuration file:

---

```
restrict default flag1 flag2 ...
```

---

If a particular source address matches more than one list entry, the entry with the most one bits in the address mask is taken to be the matched entry. If a match is found, flags associated with this entry are returned.

Suppose that you are running `xntpd` on a host with IP address 128.212.66.67. You would like to ensure that runtime reconfiguration requests can be made only from the local host. Further, you would like the host to synchronize with only one of a pair of offsite servers or, failing that, a time source on the class B network whose address is 128.212. The following entries in the configuration file would implement this policy:

---

```
By default, do not trust and do not allow modifications
restrict default notrust nomodify

These hosts are trusted for time, but no modifications allowed
restrict 128.212.0.0 mask 255.255.0.0 nomodify
restrict 128.192.8.4 nomodify
restrict 192.211.1.24 nomodify

These local addresses are unrestricted
restrict 128.212.66.67
restrict 127.0.0.1
```

---

The first entry is the default entry, which all hosts match and hence which provides the default set of flags. The next three entries indicate that matching hosts have only the `nomodify` flag set and hence are trusted for time. If the mask is not specified in the `restrict` statement, it defaults to 255.255.255.255.

Note that the address 128.212.66.67 matches three entries in the table, the default entry (mask 0.0.0.0), the entry for net 128.212 (mask 255.255.0.0) and the entry for the host itself (mask 255.255.255.255). As expected, the flags for the host are derived from the last entry, as that mask has the most bits set.

Each **restrict** statement applies to packets from all hosts, including those that are configured elsewhere in the configuration file. Hence, if you specify a default set of restrictions that you do not wish to apply to the hosts you are synchronizing with, you must override the default restrictions for those hosts with additional **restrict** statements.

### *Name resolution*

The **xntpd** daemon can specify host names requiring resolution in *peer* and *server* statements in the configuration file. This task is actually accomplished by a separate program, **xntpres**. When the daemon comes across a *peer* or *server* statement in the configuration file where the host is identified by name rather than by IP address, it records the relevant information and continues. When the end of the configuration file has been reached and one or more entries requiring name resolution have been found, **xntpres** is started. The **xntpd** daemon continues running at the same time to process those hosts with numeric addresses.

The **xntpres** program attempts to resolve each host name, meaning that it tries to obtain the IP address associated with the named host. If it succeeds in resolving the name, it reconfigures the local host to add the newly resolved host to the list of time servers to be polled. The runtime reconfiguration of the local host is accomplished using the same mode 7 runtime reconfiguration facility that **xntpd** uses. If **xntpres** is at first unable to resolve a host's name, it retries periodically until it succeeds or until it determines that the name cannot be resolved.

There are several configuration requirements if **xntpres** is to be used. The pathname of the **xntpres** program must be made known to the daemon with the **resolver** statement in the configuration file. Also, mode 7 runtime reconfiguration must be enabled with the **requestkey** statement and a list of valid keys (valid keys are specified in the *keys* file).

The following fragment might be added to */etc/ntp.conf* to accomplish this:

---

```
resolver /local/etc/xntpres
keys /etc/ntp.keys
requestkey 65535
```

---

Note that `xntpres` sends packets to the server with a source address of 127.0.0.1. If you are using the address-and-mask facility in the configuration file, you must take care not to restrict modification requests from this address or `xntpres` fails.

## Sample scenarios

---

This first scenario would be used when configuring a single host to use NTP. The host *rainbow.sci.com* is expected to run as a stratum 2 server because it is configured to get the time from two time servers with radio clocks, *bedivere.cs.purdue.edu* and *asmus1.genetics.uga.edu*. The `xntpd` daemon stores the frequency error of the clock on *rainbow.sci.com* in */etc/driftfile*. This is a good example to use if you just want to get `xntpd` up and running quickly. Note that it can take a day or two for a host to synchronize with a time server.

Here is the *ntp.conf* file for *rainbow.sci.com*.

---

```
#
Peer configuration for 128.212.66.13 (rainbow.sci.com)
#
peer 128.211.1.24 # bedivere.cs.purdue.edu
peer 128.192.8.4 # asmus1.genetics.uga.edu

driftfile /etc/driftfile
```

---

This next scenario is the complete configuration for *sci.com*. The configuration files here configure *sci.com* to match the topology discussed in “Guidelines” earlier in this chapter. All hosts can be configured dynamically with `xntpd`. There are no restrictions on which hosts can be designated as time servers in a given host’s configuration file.



## 8. Synchronizing clocks

Here is the *ntp.conf* file for Zite.

---

```
#
Peer configuration file for 128.212.66.90 (zite.sci.com)
#
peer 128.211.1.4 # percival.cs.purdue.edu
peer 192.35.53.67 # deidinator.psyc.chi.il.us
peer HGWells.Sci.com
peer DrWho.Sci.com

broadcast 128.212.66.255

driftfile /etc/ntp.drift
resolver /etc/xntpres
keys /etc/ntp.keys
requestkey 65534
controlkey 65535
```

---

Here is the *ntp.keys* file for Zite.

---

|       |   |           |
|-------|---|-----------|
| 2313  | A | APassword |
| 65534 | A | NoSecret  |
| 65535 | A | BadKey    |

---

Here is the *ntp.conf* file for HGWells.

---

```
#
Peer configuration for HGWells (128.212.66.67)
#
peer 128.192.8.4 # asmus1.genetics.uga.edu
peer 128.211.1.24 # bedivere.cs.purdue.edu
peer zite
peer DrWho

broadcast 128.212.66.255

driftfile /etc/ntp.drift
resolver /etc/xntpres
keys /etc/ntp.keys
requestkey 65534
controlkey 65535
```

---

Here is the *ntp.keys* file for *HGWells*.

---

```
2313 A APassword
65534 A NoSecret
65535 A BadKey
```

---

Here is the *ntp.conf* file for *DrWho*.

---

```
#
Peer configuration file for 128.212.66.118
DrWho.Sci.com
#
peer 192.35.54.2 # nehd.gov
peer 192.35.55.2 # berdie.athens.ga.us
peer HGWells.Sci.com
peer Zite.Sci.com

broadcast 128.212.66.255

driftfile /etc/ntp.drift
resolver /etc/xntpres
keys /etc/ntp.keys
requestkey 65534
controlkey 65535
```

---

Here is the *ntp.keys* file for *DrWho*.

---

```
2313 A APassword
65534 A NoSecret
65535 A BadKey
```

---

Here is the *ntp.conf* file for *Moondoggie*.

---

```
Peer configuration used by all stratum 3
servers at Sci.com
#
broadcastclient yes
broadcastdelay 0.0500
driftfile /etc/ntp.drift
resolver /etc/xntpres
keys /etc/ntp.keys
requestkey 65534
controlkey 65535
```

---

Here is the *ntp.keys* file for *Moondoggie*.

---

|       |   |           |
|-------|---|-----------|
| 2313  | A | APassword |
| 65534 | A | NoSecret  |
| 65535 | A | BadKey    |

---

## Testing and tuning

---

There are several parameters available for tuning time servers. These are set using the **precision** and **maxskew** configuration statements. A fragment that would simply reset these to their default values follows:

```
precision -6
maxskew 0.01
```

The **precision** statement sets the value of **sys.precision**, and the **maxskew** statement sets the value of **NTP.MAXSKW**. The variable **sys.precision** is the base two logarithm of the expected precision of the system clock. The default value is -6 on all servers.

The NTP protocol makes use of **sys.precision** in several places. For one, **sys.precision** is included in packets sent to peers and is used by them as a kind of quality indicator. When faced with selecting for synchronization purposes one of several servers, all of which are at the same stratum and offer about the same network path delay, clients prefer to synchronize to those claiming the smallest (most negative) value of **sys.precision**. The effect is particularly pronounced when all the servers are on the same LAN. Hence, if you run several stratum 1 servers, or three or four stratum 2 servers, and you would like clients to prefer one of these over the others for synchronization, you can accomplish this by decreasing the value of **sys.precision** on the preferred server or by increasing this value on the other servers, or by doing both.

The other tuning parameter is the **antihop aperture** and is derived from **sys.precision** and **NTP.MAXSKW** using the following equation:

---


$$\text{antihop aperture} = 2 ** \text{sys.precision} + \text{NTP.MAXSKW}$$


---

This equation says that the **antihop aperture** is equal to 2 raised to the **sys.precision** power plus **NTP.MAXSKW**.

Making the **antihop aperture** larger makes it less likely that the host will “hop” from the server it is currently synchronizing with to a different server. Unfortunately, this also increases the probability that the host continues to synchronize with a server whose clock is no longer accurate.

Making the **antihop aperture** smaller allows the host to hop more freely from server to server, but this can also cause it to generate a fair bit more NTP packet traffic than necessary and to no good purpose. Given the agreement among current stratum 1 NTP servers and the performance typical of the Internet, it is recommended that the **antihop aperture** be kept between 0.020 and 0.030. The default value is about 0.026.

You can change the **antihop aperture** by changing the value of **NTP.MAXSKW** using the **maxskew** configuration statement. Note, however, that if you wish to change **sys.precision** with the **precision** configuration statement, but do not wish to alter the **antihop aperture**, you must change **NTP.MAXSKW** to compensate.

## Query commands

---

Two query programs, **ntpq(ADMN)** and **xntpd(ADMN)**, are available for use by the network administrator. The **ntpq** command sends queries and receives responses using NTP standard mode 6 control messages, while the **xntpd** command uses NTP private mode 7 messages to send its requests. The format and content of these messages are specific to **xntpd**. This command allows you to inspect a wide variety of internal counters and other state data. It also provides you with an interface to the runtime reconfiguration facility.

Both **xntpd** and **xntpd** use seconds as the unit of time, both when reading the configuration file and when doing runtime reconfiguration. Both programs also print values in seconds. On the other hand, **ntpq** prints all time values in milliseconds. You must, therefore, take special care to convert values output by **ntpq** from milliseconds to seconds before modifying the configuration file, either manually or with **xntpd**.

The **xntpd** daemon is designed to allow its configuration to be modified at runtime. Nearly everything that can be configured in the configuration file **ntp.conf** may be altered using **xntpd**.

Mode 6 and mode 7 packets, which would modify the configuration of the local host, must be authenticated. To enable the facilities one must, in addition to specifying the location of a *keys* file, indicate in the configuration file the key IDs to be used for authenticating reconfiguration requests. The following fragment might be added to the configuration file to enable *ntp* and *xntpd*.

---

```
keys /etc/ntp.keys
requestkey 65535 # for mode 7 requests (used by xntpd)
controlkey 65534 # for mode 6 requests (used by ntp)
```

---

If the **requestkey** or the **controlkey** configuration statement is omitted from the configuration file, the corresponding runtime reconfiguration facility is disabled.

The query programs require the user to specify a key ID and a key to use for authenticating requests to be sent. The key ID provided must be the same one specified in the configuration file, while the key value provided must match the one corresponding to the given key ID in the *keys* file.

## Further examples

---

This example demonstrates the use of *ntp* and *xntpd* to list the servers that *Moondoggie.sci.com* is currently polling. The server marked with an asterisk "\*" is the one that *Moondoggie* is currently synchronizing with. *Moondoggie* is also listening for broadcast packets on the subnet with IP address 128.212.66, but no packets are currently being broadcast there. Note the difference in the units used by *ntp* and *xntpd* to display delay, offset, and dispersion. The former uses milliseconds, whereas the latter uses seconds.

---

```
sjc@moondoggie $ ntpq
ntp> peer
```

| remote         | refid         | st | when  | poll | reach | delay | offset | disp   |
|----------------|---------------|----|-------|------|-------|-------|--------|--------|
| 128.212.66.255 | 0.0.0.0       | 16 | never | 64   | 0     | 0.0   | 0.00   | 64000  |
| *Zite.sci.com  | deidinator.ps | 2  | 53    | 64   | 376   | 81.1  | -1.00  | 12.4   |
| +DrWho.sci.com | berdie.athens | 2  | 11    | 64   | 37    | 41.2  | 30.00  | 7504.3 |

---

```

sjc@moondoggie $ xntpd
xntpd> peer

```

| remote          | local        | st | poll | reach | delay  | offset    | disp   |
|-----------------|--------------|----|------|-------|--------|-----------|--------|
| *Zite.sci.com   | 192.35.53.67 | 2  | 64   | 376   | 0.0811 | -0.001003 | 0.0124 |
| -DrWho.sci.com  | 192.35.55.2  | 2  | 64   | 37    | 0.0412 | 0.030001  | 7.5043 |
| ^128.212.66.255 | 0.0.0.0      | 16 | 64   | 0     | 0.0000 | 0.000000  | 64.000 |

This information tells us, among other things, that:

- *Moondoggie* is currently synchronizing with *Zite*.
- *Zite* is at stratum 2, implying that *Moondoggie* is at stratum 3.
- *Moondoggie* is polling *Zite* once every 64 seconds.
- The reachability register for *Zite* is octal 376.
- The clock on *Moondoggie* is 1 millisecond, or more precisely, 1003 microseconds, behind the clock on *Zite*.
- The roundtrip delay to *Zite* is 81 milliseconds.
- The dispersion is 12.4 milliseconds.

The next example shows how to turn on monitoring and how to list the statistics that are gathered when monitoring is enabled.

```

xntpd> monitor on
Keyid: 65534
Password: NoSecret
done!

```

Now that monitoring is on, we can let the daemon run for a while, then display statistics, as shown here.

```

xntpd> monlist

```

| address        | port | count | mode | version | lasttime | firsttime |
|----------------|------|-------|------|---------|----------|-----------|
| 127.0.0.1      | 1417 | 6     | 7    | 2       | 0        | 475668    |
| 128.212.66.118 | 123  | 3818  | 1    | 2       | 92       | 475332    |
| 128.212.66.90  | 123  | 7432  | 5    | 2       | 48       | 475632    |

This display tells us who has connected to the `xntpd` daemon, on what UDP port the connection was made, the number of packets received, the mode of this connection, the version of NTP being run, the time since the last packet exchange in seconds, and the time since the first packet exchange in seconds.

The last example shows how we can dynamically delete and add time servers. First, we indicate which host we wish to communicate with, then we display the current peers of that host.

---

```
xntpd> host rainbow
current host set to rainbow.sci.com

xntpd> peers
```

| remote           | local       | st | poll | reach | delay  | offset    | disp   |
|------------------|-------------|----|------|-------|--------|-----------|--------|
| =====            |             |    |      |       |        |           |        |
| *zite.sci.com    | 128.211.1.4 | 2  | 64   | 376   | 0.0512 | -0.010000 | 0.0021 |
| +HGWells.sci.com | 0.0.0.0     | 16 | 1024 | 0     | 0.0000 | 0.000000  | 64.000 |

---

Now we remove *HGWells* as a peer and view the result.

---

```
xntpd> unconfig HGWells
Keyid: 65534
Password: NoSecret
done!

xntpd> peers
```

| remote        | local       | st | poll | reach | delay  | offset    | disp   |
|---------------|-------------|----|------|-------|--------|-----------|--------|
| =====         |             |    |      |       |        |           |        |
| *zite.sci.com | 128.211.1.4 | 2  | 64   | 376   | 0.0512 | -0.010000 | 0.0021 |

---

Now that *HGWells* has been removed as a peer, we go ahead and add host *DrWho* as a new peer.

---

```
xntpd> addpeer DrWho.sci.com
done!
```

```
xntpd> peers
```

| remote         | local       | st | poll | reach | delay  | offset    | disp   |
|----------------|-------------|----|------|-------|--------|-----------|--------|
| =====          |             |    |      |       |        |           |        |
| +DrWho.sci.com | 0.0.0.0     | 16 | 64   | 0     | 0.0000 | 0.000000  | 64.000 |
| *zite.sci.com  | 128.211.1.4 | 2  | 64   | 376   | 0.0512 | -0.010000 | 0.0021 |

---

If we allow *Rainbow* to poll the new peer for a period of time and then reissue the **peers** subcommand, we might see something like this display.

---

```
xntpd> peers
```

| remote         | local          | st | poll | reach | delay  | offset    | disp   |
|----------------|----------------|----|------|-------|--------|-----------|--------|
| =====          |                |    |      |       |        |           |        |
| *DrWho.sci.com | 128.212.66.118 | 2  | 64   | 377   | 0.0312 | -0.004488 | 0.0011 |
| ~zite.sci.com  | 128.211.1.4    | 2  | 64   | 376   | 0.0512 | -0.010000 | 0.0021 |

---

## Troubleshooting

---

The **xntpd** daemon records all problems and errors using the **syslog**(SLIB) system call into the file */usr/adm/syslog*. You can examine this log for signs of trouble. For example, too many entries indicating that the local clock was stepped instead of slewed to the correct time suggests that the daemon is having trouble synchronizing the clock. (In this instance, more than two entries per hour on a consistent basis is probably too many.) The most likely cause is that some other program, perhaps **timed**, is also setting or slewing the clock.

Another sign of more than one entity setting or slewing the clock is the following message in the *syslog* file:

---

```
Previous time adjustment did not complete
```

---



## 8. Synchronizing clocks

One of the following messages is issued when the clock of another host (the remote clock) is off by more than 1000 seconds from the clock on the local host (the local clock):

---

Clock appears to be *number\_of\_seconds* seconds fast,  
something may be wrong

---

---

Clock appears to be *number\_of\_seconds* seconds slow,  
something may be wrong

---

In this situation `xntpd` will not try to synchronize with that host. If you want to synchronize with that host, you will need to set the local clock to within 1000 seconds of the remote clock. You can accomplish this by using the `ntpdate(ADMN)` command.

### *Running mixed synchronization subnets*

---

The `xntpd` time daemon is an implementation of NTP Version 2. By default, `xntpd` sends NTP packets marked as being of the Version 2 variety. However, another implementation of the NTP time daemon, called `ntpd`, may be running on hosts you wish to poll. If this is the case, you must include the `version` option in the `peer` or `server` configuration statements for those hosts to force `xntpd` to send packets of the Version 1 variety.

For example, if the host `nehd.gov` at IP address 192.35.54.2 is a host you wish to poll in symmetric active mode and you know that `nehd.gov` is running `ntpd` instead of `xntpd`, you should include the following configuration statement in your local hosts `ntp.conf` file:

---

```
peer 192.35.54.2 version 1 # nehd.gov (running ntpd)
```

---

This statement ensures that Version 1 NTP packets are sent by your local host to `nehd.gov`.

## Chapter 9

# Administering NFS

---

In NFS, a server exports directories to be shared and clients mount the directories to access the files in them. The following subsections describe the general interaction of files and daemons in these activities. For details of what happens when a client mounts a remote filesystem, see the section “Remote mount failed” later in this chapter.

## Basic concepts

---

The following sections explain the basic functionality of NFS and the daemons that provide it.

### How files interact

---

To export a filesystem, the server must list it along with access rights in its */etc/exports* file. In addition, the server must have a name and address for each client to receive services in its */etc/hosts* file.

To mount a directory, the client simply issues a **mount** command, specifying the filesystem to be mounted, the filesystem’s server, and the local mount point. The **mount(NADM)** command automatically adds an entry to the client’s */etc/mnttab(F)* file; the **umount** command automatically deletes the entry in this file. Consequently, this file maintains a current list of mounted filesystems.

The client can also mount remote filesystems by default when the system starts by placing an entry in */etc/default/filesys*. The client’s */etc/default/filesys* file must list all remote filesystems for which access is desired and their locations.

**NOTE** NFS supports resource sharing by exporting filesystems from servers. However, the applications that access these shared filesystems reside on the client host and use the client's CPU and RAM. Therefore, filesystems are distributed in NFS, not applications.

### *How daemons interact*

---

Two remote programs provide the NFS service — **mountd**(NADM) and **nfstd**(NADM). A client's **mount** request calls **mountd**, which checks the server's */etc/exports* file for access permission of the client and returns a pointer to a filesystem. After **mount** completes, access to that mount point directory and below goes through the **nfstd** daemon on the server system. Client-kernel file access requests (delayed-write and read-ahead) are handled by the **biod** (see **nfstd**(NADM)) daemons on the client.

NFS client requests are fulfilled through Remote Procedure Calls (RPC). Using RPC, the client process can cause the server process to execute a procedure call, as if the client process had executed the procedure call in its own address space. The **portmap**(NADM) daemon maps RPC procedure calls to the appropriate TCP/IP port numbers (that is, to these calling processes) and thus is also an essential part of the NFS service.

**pcnfsd** is used by machines running the DOS version of NFS. For further information, refer to the **pcnfsd**(NADM) manual page.

### *Managing user accounts*

---

Within a network environment, each user and group must have a unique identification number across the entire network, rather than just on the home machine. The user ID should be a number unique to the user. A system knows the user by the ID number associated with the login name; therefore, a login name must have the same user ID number in all password files of machines that are networked in a local domain.

**CAUTION** Failure to keep IDs unique prevents users from moving files between directories on different machines, because the system responds as if the directories were owned by two different users. In addition, file ownership may become confused when an NFS server exports a directory to an NFS client whose password file contains users with UIDs matching those of different users on the NFS server.

If you wish to maintain user equivalence, each new user must be added to the proper password files for all machines on the system. For general information on how to do this, see the *Administering Operating System Services* guide. If you are using NIS, see also the "NIS: Network Information Service" chapter of this guide.

# Configuring NFS

Preparing NFS for use across a network entails setting up NFS servers and clients, mounting remote filesystems, and starting the NFS daemons. The following sections explain these operations.

## Setting up an NFS server

Setting up an NFS server involves enabling a machine to export a filesystem and checking the `/etc/hosts` file. All examples in this section use the server named *jupiter*.

### Enabling export

Follow these steps to enable a machine to export a filesystem.

1. Log in as *root*.
2. Change (`cd`) to the `/etc` directory. Edit the `/etc/exports` file using your preferred editor to add the mount-point pathname of the filesystem to be exported. For example:

```
/usr/man # export to any machine
/usr2 -access=venus:mars # export to only these machines
```

See the `exports(NF)` manual page for syntax and contents of this file.

**NOTE** An NFS server can only export filesystems of its own. For example, if a client *mars* remote-mounted the filesystem `/tools` from the server *jupiter*, *mars* cannot in turn export `/tools` to a third client. Thus, NFS does not support “daisy-chaining” (third-party mounting).

3. Make sure `/etc/mountd` is running. This daemon is started from the NFS startup script, usually `/etc/nfs`, and must be running for a remote mount to succeed. Check `/etc/nfs` for lines such as the following to enable `mountd`:

```
if [-r /etc/exports]
then
 echo " mountd\c"
 mountd
fi
```

4. Make sure some number of **nfsd** NFS daemon processes are running on the NFS server. The actual number depends on the number of client NFS requests that the server should be able to handle concurrently, and thus depends on the speed and capacity of the server machine.

Check the NFS startup script, */etc/nfs*, for lines like these to enable **nfsd**:

```
if [-r /etc/exports]
then
 echo " nfsd(x4)\c"
 nfsd 4
fi
```

This example invokes four **nfsd** daemons.

The super user can enable **nfsd** at any time by typing:

```
/etc/nfsd 4
```

5. Export the files in */etc/exports* by entering the command:

```
/etc/exportfs -a
```

Once the server is enabled, **exportfs** is run as part of the startup script whenever NFS is started. See the **exportfs(NADM)** manual page for more information,

After these steps, the NFS server should be able to export the named file-system.

**NOTE** Because of potential problems with user equivalence, if you intend to export any user's **HOME** directory you must remember to reset the permissions on that directory so that the directory is accessible for its intended use (default permissions are set to 0700). For information on how to reset permissions see, **chmod((C))**.

### *Checking the **etc/hosts** file*

The name, network address, and aliases of each NFS client or server host must appear in the */etc/hosts* file. If this information is not present, log in as the super user and edit the file.

Here is a sample file:

```
102.6.3.76 jupiter
192.9.1.20 earth
176.2.5.42 venus
142.9.7.36 mars
```

See the **hosts(SFF)** manual page for the syntax and contents of the file.

## Setting up an NFS client

To enable an NFS client, you need only mount a remote filesystem after NFS is started. You can do so by issuing the `mount` command or by specifying remote filesystems in `/etc/default/filesys` to be mounted by default whenever NFS is started.

### Mounting the remote filesystem

You can remote-mount any exported filesystem onto your client machine, as long as the filesystem's server can be reached over the network and the client machine is included in the server's `/etc/exports` list for that filesystem. On the machine where the filesystem is to be mounted, the super user executes the `mount` command. For example, to mount the manual pages from remote machine *jupiter* on the directory `/usr/jupiter.man`, enter:

```
mount -r -f NFS,soft,bg jupiter:/usr/man /usr/jupiter.man
```

This example illustrates the use of the `soft` option to specify a soft mount, which means that the client does not retry the mount operation if the server does not respond to the first request. The default is a hard mount, which means that the client continues to try to mount the requested directory even if the server does not respond. A soft mount is recommended for read-only directories, such as online manual pages. The underlying reasoning is that with a soft mount, halting the server does not halt the client.

This example also illustrates the use of the `bg` to specify that the mount take place in the background. The default specifies foreground mounting; however, if the server is not responding, the client hangs while waiting for the server to respond. Background mounting is recommended for the same reason as soft mounting: to protect the client if the server is not available.

To make sure the filesystem is mounted where it is expected to be, use the `mount` command without any arguments. This displays the currently mounted filesystems.

### Enabling default mounting

If you mount certain remote filesystems regularly, you can specify them for default mounting in `/etc/default/filesys`. If a filesystem has an entry in `/etc/default/filesys`, it is mounted whenever the client is rebooted. A client uses its `/etc/default/filesys` file to keep track of specific remote filesystems and directory locations, called *mount points*, through which the client can access directories.

Make sure that all filesystem information, including access rights, appears in this file. If any information is not present, edit the file to add it. In addition, use `mkdir` to create any mount points that do not already exist in the filesystem.

Here is a sample entry:

```
bdev=jupiter:/usr/man mountdir=/usr/man \
 fsck=no rcfck=no rcmount=yes \
 mount=no fstyp=NFS \
 nfsops="soft,bg,intr,rsiz=1024,wsiz=1024"
```

Table 9-1 explains the entry.

**Table 9-1** */etc/default/filesys* format

| Entry    | Value                   | Indicates                                                                                                                  |
|----------|-------------------------|----------------------------------------------------------------------------------------------------------------------------|
| bdev     | <i>jupiter:/usr/man</i> | the remote filesystem to mount                                                                                             |
| mountdir | <i>/usr/man</i>         | where to mount the filesystem                                                                                              |
| fsck     | no                      | whether to check the filesystem with fsck(ADM) when a user mounts the filesystem with mnt                                  |
| rcfsck   | no                      | whether to check the filesystem with fsck(ADM) when the system enters multiuser mode                                       |
| rcmount  | yes                     | whether the filesystem is automatically mounted when the system enters multiuser mode                                      |
| mount    | no                      | whether users can mount and unmount this filesystem                                                                        |
| fstyp    | NFS                     | the filesystem type; this must always be set to NFS for SCO NFS to function correctly                                      |
| nfsops   | soft                    | for recovery after server failure; recommended only for read-only filesystems                                              |
|          | bg                      | so the mount takes place in the background; recommended to prevent the client from hanging if the server is not responding |
|          | intr                    | so interrupts work across the network                                                                                      |

If you have a 3Com EtherLink 3C501 card on the client or server, you must set the **rsiz** and **wsiz** parameters in **nfsops** as shown in the sample entry to prevent excessive NFS retransmissions.

See the *filesys(F)* manual page for a complete description of the syntax and contents of */etc/default/filesys*.

**NOTE** The **mkdev fs** command cannot be used to add NFS entries to */etc/default/filesys*.

## Enabling user mounting

Under normal circumstances, only the super-user can mount filesystems using NFS. However, you can enable users to mount and unmount selected filesystems using the `mnt(C)` and `umnt(C)` commands. Enabling user mounting can reduce the amount of time it takes to bring your system into multiuser mode and reduce the amount of network traffic by ensuring that filesystems are mounted only when needed.

To enable user mounting, you must create an entry in `/etc/default/filesys` for each filesystem to be mounted. The syntax that enables user mounting is the same as for default mounting (listed in the previous section, “Enabling default mounting”) except for two parameters:

`mount=yes` enables users to mount and unmount the filesystem  
`rcmount=no` prevents the filesystem from being mounted automatically when the system enters multiuser mode

The following sample entry allows a user to mount the `/usr/man` directory from the server `jupiter` (if `/usr/man` is exported by `jupiter`):

```
bdev=jupiter:/usr/man mountdir=/usr/man \
 fsck=no rcfscck=no rcmount=no \
 mount=yes fstyp=NFS \
 nfsopts="soft,bg,intr,rsiz=1024,wsiz=1024"
```

The user can mount this directory by issuing the command `mnt` with the mountpoint listed in the `mountdir` parameter, for example:

```
mnt /usr/man
```

The filesystem can be unmounted using the `umnt` command with the mount point as argument.

You should instruct your users in proper mounting procedures:

- Filesystems should be unmounted when not in regular use.
- The `mnt` and `umnt` commands cannot be issued if the user's current working directory is the mount point listed in `/etc/default/filesys`. Otherwise, an error message `Mount point busy` will be returned.
- Users can display a list of mounted filesystems by issuing the `mnt` command without an argument. The command `mnt -t` will list the contents of `/etc/default/filesys`; users are unable to mount any file system if the value given for the line “Mount if requested” is “No.”

Refer to the `mnt(C)` manual page for more information.



**NOTE** You can also enable user mounting transparently by using Automounter. If you create an Automounter map for an exported filesystem rather than an entry in */etc/default/filesys*, the filesystem will be mounted automatically whenever the filesystem or one of its files are accessed. Such a mount is transparent to the user, and the filesystem unmounts automatically after a certain period of inactivity. For more information, see the “Automounter” chapter of this guide.

### *Starting and stopping NFS*

---

NFS must be started by a shell script invoked by *init*(M) when the system is brought up to multiuser mode (run level 2). This is normally configured automatically during installation by linking the script */etc/nfs* to */etc/rc2.d/S89nfs*. You must start NFS on at least one server and at least one client to have a functional NFS network.

To stop NFS on a machine, log in as *root* and enter the following command:

```
/etc/nfs stop
```

When you stop NFS on a node, its resources cannot be accessed from any other node. NFS is shut down automatically when the system is taken to run levels 0, 1, or 6.

## *Administering NFS*

---

The following sections explain procedures for NFS debugging and maintenance.

### *Debugging NFS*

---

Most problems involving System V NFS network services lie in one of the following four areas, which are listed in order of probability:

- The network-access control policies do not allow the operation, or architectural constraints prevent it.
- The client software or environment is not functioning correctly.
- The server software or environment is not functioning correctly.
- The network is not functioning correctly.

To debug NFS, you should be familiar with how NFS works and the names and functions of the various daemons and database files:

|                                      |                                           |
|--------------------------------------|-------------------------------------------|
| <b>biod</b> (see <b>nfsd</b> (NADM)) | NFS client I/O daemon                     |
| <b>mount</b> (NADM)                  | mounts a filesystem                       |
| <b>mountd</b> (NADM)                 | NFS mount request server                  |
| <b>nfsd</b> (NADM)                   | NFS server I/O daemon                     |
| <b>rpcinfo</b> (NADM)                | reports RPC information                   |
| <b>showmount</b> (NADM)              | shows all clients with remote mounts      |
| <b>filesystem</b> (F)                | format of filesystem mount commands table |
| <b>mnttab</b> (NF)                   | format of mounted filesystem table        |
| <b>exports</b> (NF)                  | NFS filesystems being exported            |

For example, consider a sample mount request as made from an NFS client machine:

```
mount -f NFS krypton:/usr/src /krypton.src
```

The example asks the server machine *krypton* to return a file handle (**fhandle**) for the directory */usr/src*. This **fhandle** is then passed to the kernel in the **mount**(NS) system call. The kernel looks up the directory */krypton.src* and, if everything is correct, ties the **fhandle** to the directory in a mount record. From now on, all filesystem requests to that directory and below go through the **fhandle** to the server *krypton*.

This describes the way the system should work. We now look at the things that can go wrong: first, some general pointers for debugging NFS; then, a list of specific errors and their probable causes.

## General hints

When there are network or server problems, programs that access hard-mounted remote files fail in different ways from those that access soft-mounted remote files. Hard-mounted remote filesystems cause programs to retry until the server responds again, while soft-mounted remote filesystems return errors after a certain number of retries.

Once a hard mount succeeds, programs that access hard-mounted files hang as long as the server fails to respond. In this case, NFS displays the following message on the console:

```
server_name not responding
```

On a soft-mounted filesystem, a program receives an error message when it tries to access a file whose server is not responding. You may also wish to specify the **intr** option when hard-mounting filesystems from an unreliable server; this allows you to interrupt the **mount** command and returns an error message from the server.

If a client is having NFS trouble, first check that the server is up and running. From a client, type the following command:

```
rpcinfo -p server_name
```

It should display a list of program, version, protocol, and port numbers similar to the following:

| program | vers | proto | port |            |
|---------|------|-------|------|------------|
| 100000  | 2    | tcp   | 111  | portmapper |
| 100000  | 2    | udp   | 111  | portmapper |
| 100017  | 1    | tcp   | 1024 | rexed      |
| 100005  | 1    | udp   | 1027 | mountd     |
| 100003  | 2    | udp   | 2049 | nfs        |
| 100024  | 1    | udp   | 1039 | status     |
| 100024  | 1    | tcp   | 1025 | status     |
| 100021  | 1    | tcp   | 1026 | nlockmgr   |
| 100021  | 1    | udp   | 1051 | nlockmgr   |
| 100020  | 1    | udp   | 1054 | llockmgr   |
| 100020  | 1    | tcp   | 1028 | llockmgr   |
| 100021  | 2    | tcp   | 1032 | nlockmgr   |

You can also use **rpcinfo** to check if the **mountd** server is running:

```
rpcinfo -u server_name mountd
```

This should return:

```
program 100005 version 1 ready and waiting
```

If these steps fail, run the **rpcinfo -p** command on the server's console to verify that the programs listed above are registered.

If the server is functional but a client machine cannot reach it, check the Ethernet connections between the machines.

If the server and the network are functional, use **ps** to check the client daemons. A **nfscnt** and several **biod** daemons should be running. For example, the command **ps -ef** should display lines for **nfscnt** and **biod**.

The following sections deal with the most common types of failure. "Remote mount failed" covers the steps to take if a remote mount fails. The sections "Programs are hung" and "Everything works slowly" discuss servers that do not respond when filesystems are mounted.

### *Remote mount failed*

If **mount** fails for any reason, check the sections below for specific details about what to do. They are arranged according to where they occur in the mounting sequence and labeled with the error message likely to be displayed.

The **mount** command can get its parameters either from the command line or from the file */etc/default/filesys* (see the *mount(NADM)* manual page for further

information). The following example assumes command-line arguments, but the same debugging techniques would apply if */etc/default/filesys* were the source of the options.

```
mount -f NFS krypton:/usr/src /krypton.src
```

The interaction of the various parts of the mount request should be considered. In the preceding example, **mount** goes through the following steps to mount a remote filesystem:

1. The **mount** command opens */etc/mnttab* and checks that this mount was not already done.
2. The **mount** command parses the first argument into host *krypton* and remote directory */usr/src*.
3. The **mount** command then resolves the host *krypton* into an Internet protocol address.
4. The **mount** command calls *krypton*'s portmapper to get the port number of **mountd**.
5. The **mount** command calls *krypton*'s **mountd** and passes it the */usr/src* pathname.
6. *krypton*'s **mountd** reads its */etc/exports* and looks for the exported filesystem that contains */usr/src*.
7. *krypton*'s **mountd** does a **nfs\_getfh(NS)** system call on */usr/src* to get the **fhandle**.
8. *krypton*'s **mountd** returns the **fhandle** to the client.
9. On the client machine, **mount** does a **mount(NS)** system call with the **fhandle** and */krypton.src*.
10. The **mount** command checks whether or not the caller is a super user and */krypton.src* is a directory.
11. The **mount** command does a **statfs(S)** call to *krypton*'s NFS server (**nfsd**).
12. The **mount** command opens */etc/mnttab* and adds an entry.

Any one of these steps can fail, and some of them can fail in more than one way. The following sections give detailed descriptions of the failures associated with specific error messages.

```
mount: can't open /etc/mnttab
```

The table of mounted filesystems is kept in the file */etc/mnttab*. This file must exist before **mount** can succeed. The file */etc/mnttab* is created when the system is booted, and it is maintained automatically after that by the **mount** and **umount** commands.

mount: /dev/nfsd is already mounted, *sys\_name* is busy, or allowable number of mount points exceeded

This message reveals an attempt to mount a filesystem that is already mounted. All NFS mount requests that fail with this message display the name */dev/nfsd* (a byproduct of the implementation) regardless of the actual mount request.

mount: *name* or *name*, no such file or directory

The **-f** NFS or **krypton:** part of the sample command was probably omitted. The **mount** command assumes that a local mount is being done unless the **-f** flag is used on the command line, or the requested directory as listed in */etc/default/filesys* specifies filesystem type NFS.

More simply, this message also appears when the specified local mount point is not an existing directory.

mount: can't open /etc/default/filesys

The **mount** command tried to look up the information needed to complete a mount request in */etc/default/filesys*, but there was no such file. As the system administrator, you need to create this file as part of initial system setup.

*sys\_name* not in hosts database

The system name specified on the **mount** command suffixed by the **":"** is not listed in the file */etc/hosts*. Check the spelling of the host name and the placement of the colon in the **mount** command.

mount: directory argument *name* must be a full pathname

The second argument to **mount** is the path of the directory to be covered. This must be an absolute path starting at slash **/**.

mount: ... server not responding(1):  
RPC\_PMAP\_FAILURE - RPC\_TIMED\_OUT

Either the server to which the mount is being attempted is down, or its portmapper is dead or hung. Attempt to log in to that machine; if you succeed, then the problem may be in the portmapper. Run the following command from your system as the super user to test the portmapper on the server system:

**rpcinfo -p host\_name**

The result should be a list of registered programs. If this is not the case, kill the remote portmapper and restart it. Restarting the portmapper is a complicated process because all registered services are lost, and their associated daemons must be restarted, too. To find the process IDs of **portmap** and other service daemons, enter this command as the super user:

```
ps -ef
```

Kill the daemons with:

```
kill -9 portmap_pid daemon_id1 daemon_id2
```

Start new ones with commands such as:

```
/etc/portmap
/etc/mountd
/etc/nfsd
```

Another alternative to all this is simply to reboot the server when it is convenient. Because of the stateless nature of the NFS server implementation, there should be no adverse effect on the clients of the system, other than the time that they suspend awaiting the return of the server.

If the server is up but it is not possible to reach it using **ping**, check the client's Ethernet connection by trying to **ping** another machine. Also, check the server's Ethernet connection.

```
mount: ... server not responding:
RPC_PROG_NOT_REGISTERED
```

This means that **mount** got through to the portmapper, but the NFS mount daemon **mountd** was not registered. The server should be checked to ensure that */etc/mountd* exists and is running.

```
mount: /dev/nfsd or directory_name, no such file or directory
```

Either the remote directory does not exist on the server or the local directory does not exist. Again, note that */dev/nfsd* is always printed to represent the remote directory.

```
mount: access denied for sys_name:name
```

The client machine on which the mount attempt is being made is not in the server's export list for the filesystem to be mounted. A list of the server's exported filesystems can be obtained by running:

```
showmount -e host_name
```

If the desired filesystem is not in the list, or the machine name is not in the user list for the filesystem, then check the */etc/exports* file on the server for the correct filesystem entry. A filesystem name that appears in the */etc/exports* file but not in the output from `showmount` indicates a failure in `mountd`. Perhaps it could not read that line in the file, or it could not find the filesystem, or else the filesystem name was not a locally mounted filesystem. See `exports(NF)` for more information.

This message can also be an indication that authentication failed on the server. It may be displayed because the machine that is attempting the mount is not in the server's export list; check the server's */etc/exports* file.

mount: **name**: no such file or directory

The remote path on the server is not a directory.

mount: not super user

The `mount` command can be used only by the super user, because it affects the filesystem for the entire machine.

### *Programs are hung*

If programs hang doing file-related work, the NFS server may have crashed. The following message may appear on the machine's console:

NFS server *sys\_name* not responding, still trying

The message includes the name of the NFS server that is down. This is probably a problem either with one of the NFS servers or with the Ethernet.

If a client machine completely stops responding, check the servers from which the filesystems were mounted. If one of them is down, client machines may wait indefinitely. When the server comes back up, programs continue automatically and are unaffected.

If a soft-mounted server dies, other work should not be affected. Programs that time-out trying to access soft-mounted remote files will fail, but it should still be possible to get work done on other filesystems.

## *Everything works slowly*

If access to remote files seems unusually slow, the server should be checked by entering (on the server):

```
ps -ef more
```

If the server is functioning and other users are getting good response, use the `ps -ef` command on the client to see if block I/O (`biod`) daemons are running. If the daemons are not running or are hung, they should be killed. First, find the process IDs by entering:

```
ps -ef | grep biod
```

Then, kill them with:

```
kill -9 pid1 pid2 pid3 pid4
```

Restart the daemons with:

```
/etc/biod 4
```

To determine whether the daemons are hung, use `ps` as above, then copy a large file. Another `ps` shows whether the `biods` are accumulating CPU time; if not, they are probably hung.

If `biod` appears to be functioning correctly, check the Ethernet connection. Use the `nfsstat -c` and `nfsstat -s` commands to discover whether a client is doing a lot of retransmitting. A retransmission rate of 5% is considered high. Excessive retransmission usually indicates a bad Ethernet board, a bad Ethernet tap, a mismatch between board and tap, or a mismatch between the client machine's Ethernet board and the server's board. Insufficient STREAMS buffers may also cause excessive retransmissions. Refer to "STREAMS Tuning" in Chapter 2 of this guide.

## *Serial number validation errors*

When you see a message of the following nature, a validation error has occurred:

```
Copy Protection Violation!
Duplicate Serial Number(s) detected on 123.4.567.007
Product(s) att00001 shutting down.
```

This means that there are two copies of the same program running at the same time. Contact your SCO representative to obtain a valid activation key for the second copy.



## ***Incompatibilities with remote filesystems***

---

A few things work in unexpected ways, or not at all, on remote NFS filesystems. The next section discusses the incompatibilities and offers suggestions for working around them.

### ***File operations not supported***

File locking operations on directories is not supported on remote filesystems.

Append mode and atomic writes are not guaranteed to work on remote files accessed by more than one client simultaneously.

Applications that use mandatory locking are not supported over NFS, nor are applications that use `read()`. An application that uses the `read()` function returns an error message similar to the following:

```
25 Apr 1990 rfs_read: attempt to read from non-file
```

### ***Cannot access remote devices***

With NFS, it is not possible to access a remote-mounted device or any other character, block-special file, or named pipes.

### ***Cannot access different filesystem types***

Under NFS, it is not possible to access a filesystem of a type different from that of your native system. For example, if you are running NFS under the UNIX system, it is not possible to access a DOS filesystem from your UNIX client machine. SCO NFS only supports Acer, S51K, and UNIX filesystems.

### ***Cannot access indirect filesystems***

Under NFS, it is not possible to access a filesystem indirectly. For example, you may not use a server machine to access a filesystem on a third server machine. All NFS access must be direct from server to client.

### ***Only 8 groups per user supported when mounting a Release 1.1.1 filesystem***

To mount an SCO NFS 1.1.1 filesystem (which supports only 8 supplementary groups per user) on your SCO Open Desktop/SCO Open Server Release 2.0 or later system (which supports 16 supplementary groups per user), you will have to ensure that the user who is performing the mount is a member of no more than eight groups, as defined by the `/etc/group` file.

## Handling clock skew in user programs

Because the NFS architecture differs in some minor ways from earlier versions of the XENIX and UNIX systems, users should be aware of those places where their own programs could run up against these incompatibilities.

The clocks on the NFS server and client may be out of sync because each machine keeps its own time. This might cause problems. Consider the following example.

Many programs assume that an existing file could not be created in the future. For example, the command `ls -l` has two basic forms of output, depending upon how old the file is:

```
$ date
Sat Apr 12 15:27:48 GMT 1986
$ touch file2
$ ls -l file*
-rw-r--r-- 1 jsbach 0 Dec 27 1984 file
-rw-r--r-- 1 jsbach 0 Apr 12 15:27 file2
```

The first type of output from `ls` prints the year, month, and day of last file modification if the file is more than six months old. The second form prints the month, day, hour, and minute of last file modification if the file is less than six months old.

The `ls` command calculates the age of a file simply by subtracting the modification time of the file from the current time. If the result is greater than six months, the file is "old".

Assume that the time on the server is Apr 12 15:30:31, which is three minutes ahead of the local machine's time:

```
$ date
Apr 12 15:27:31 GMT 1986
$ touch file3
$ ls -l file*
-rw-r--r-- 1 jsbach 0 Dec 27 1983 file
-rw-r--r-- 1 jsbach 0 Apr 12 15:26 file2
-rw-r--r-- 1 jsbach 0 Apr 12 1986 file3
```

The difference between the current time and the library's modify time makes the `ls` command think that the new file was created long ago.

In general, users should remember that applications that depend upon local time and/or the filesystem timestamps have to deal with clock skew problems if remote files are used.

## Chapter 10

# NIS Network Information Service

---

Network Information Service (NIS) is a networked system administration service that supplements NFS (Network File System). This chapter presents background information on NIS, suggests guidelines for planning an NIS configuration, and explains configuration and administration procedures. Novice and experienced network administrators alike should read the entire chapter; although you may be familiar with NIS functionality, the section “Understanding NIS” includes details on features unique to SCO’s implementation.

The term “Network Information Service (NIS)” now refers to the service formerly known as Sun “Yellow Pages” or “YP.” The functionality remains the same; only the name has changed. The name “Yellow Pages” is a registered trademark in the United Kingdom of British Telecommunications PLC and may not be used without permission from British Telecom. Nonetheless, many NIS utilities retain the “yp-” prefix, for example, `ypmake` and `ypserv`.

## Understanding NIS

---

The primary purpose of NIS is to ensure the consistency of system administration information in a heterogeneous network environment. It does so by designating a single host as the *master* of all the files and databases containing this information and providing a method to distribute the information to all other hosts on the network.

NIS distributes information from the following files on the master server by default:

- */etc/passwd*
- */etc/ethers*
- */etc/networks*
- */etc/services*
- */etc/hosts*
- */etc/group*
- */etc/protocols*

*/usr/lib/mail/aliases* can also be managed by NIS if your system uses **sendmail**. In addition, other frequently used files (such as telephone lists and Auto-mounter maps) may be selected for NIS administration. When NIS is initialized on the master server, information from these files is compiled into a series of databases called *maps*. The maps are then propagated to other servers (*slave* and *copy-only* servers) on the network, where they are translated back into ASCII files. When it is necessary to update information in these files, the ASCII files are edited on the master server, and the maps are modified and repropagated using NIS commands.

In the SCO implementation of NIS, all look-ups of NIS-managed information are performed locally on servers. When a new map is received, it is converted to the corresponding ASCII file(s) that are then available locally to requesting processes. Hence, there are no clients in the SCO NIS implementation; that is, requests are not satisfied remotely, across the network. However, SCO NIS servers can satisfy look-up requests from non-SCO clients.

The following sections describe the basic concepts and functionality of NIS.

### *NIS maps*

---

NIS serves information stored in NIS *maps*, each having its own name. Maps are a series of databases containing a set of keys and associated values. For example, in a map called *hosts*, all of the host names within a network are the keys, and the Internet addresses of these host names are the values. Programs may use the *mapname* to access the map and must know the format of the data in the map.

Many of the current maps are derived from ASCII files traditionally found in */etc*, such as *hosts*, *group*, *passwd*, and others. In most cases, the data within the NIS map is identical to the format within the ASCII file. Maps are implemented as files located in the subdirectories of the directory */etc/yp* on NIS server machines.

Each map consists of a pair of *dbm*(S) database files, one containing a directory (that is, a bitmap of indices accessed by a hash table) and the other data. *dbm* files are created when NIS is initialized or as needed using NIS commands. Because of the non-ASCII file structure imposed by *dbm*, you must use NIS tools (such as *ypxfr* and *yppush*) to transfer maps from one machine to another.

## NIS domain

---

An NIS *domain* consists of a group of computers sharing configuration information on a network, although NIS domains differ from Internet, MMDF, and **sendmail** domains. Each NIS domain has a corresponding set of maps that are contained in a subdirectory of */etc/yp*; the name of the subdirectory is the same as the domain. For example, maps for the literature domain are in */etc/yp/literature*.

To retrieve data from an NIS database, you must set an NIS domain name with the **domainname(NC)** command. It is easier to manage your network if you use a single domain name on all machines on a network.

## NIS servers

---

Three kinds of machines have NIS databases in an SCO system: a *master* server, a *slave* server, and a *copy-only* server. The master server updates the databases of the other two servers. Both slave and copy-only servers handle maps, setup, and administration in a similar way. The only difference is that a slave server may satisfy requests to transfer maps if the master server becomes unavailable. In a heterogenous environment, a slave server may also accept NIS requests from any non-SCO client, but the copy-only server does not accept NIS requests.

**NOTE** The copy-only server is not a client in the conventional sense of the term. That is, an NIS request entered from a copy-only server machine is not fulfilled over the network but rather on the local host. SCO NFS does not provide NIS client functionality, but non-SCO NIS clients may make requests of SCO NIS master and slave servers.

Database changes in an NIS domain are propagated from the master server to other servers. If you create or change NIS databases on slave server or copy-only server machines instead of master server machines, all modifications you make are only temporary and are overwritten when the master server updates them. Therefore, make all database creations and modifications on the master server machine only.

A server can be a master with regard to one NIS map and a slave or copy-only with regard to another. To avoid the confusion caused by randomly assigning maps to server and copy-only machines, we recommend that you make a single server the master for all the maps created by **ypinit(NADM)** within a single domain. This document assumes that one server is the master for all maps in the database.

## NIS daemons

---

NIS services are implemented by the following daemons:

- |                 |                                                                                                                                                                                                                                                                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ypserv</b>   | satisfies requests for NIS services. Requests are usually to transfer a map, display its contents, or to display information about the NIS domain.                                                                                                                                                                                                    |
| <b>ypbind</b>   | listens for NIS requests and directs them to a <b>ypserv</b> process that can satisfy the request                                                                                                                                                                                                                                                     |
| <b>ypxfrd</b>   | satisfies requests from slave or copy-only servers to transfer maps. In SCO NIS, maps are transferred only when the identity of the master has been verified by the slave or copy-only server and a <b>ypxfr</b> command is issued.                                                                                                                   |
| <b>yppasswd</b> | satisfies requests to change passwords of NIS-managed accounts and communicates with the master server to update password information. This service is not available to servers in Secure Mode (see the "Security limitations on configuration" section of this chapter for more information). <b>yppasswd</b> must only be run on the master server. |

The action of associating a server with a particular host is called *binding* that host in the NIS domain. Hosts must be *bound* in order for NIS services to be provided. Binding is accomplished at system start-up when the **ypbind** daemon broadcasts for a server to satisfy a process request. Unless the host system crashes or the network administrator intervenes, bindings remain constant in an NIS domain.

A heterogenous environment differs from an SCO environment with regard to binding:

- SCO environment
  - all hosts are servers that contain both NIS maps and ASCII files derived from them
  - look-ups are resolved locally
  - all hosts are bound to themselves by default. That is, the local **ypbind** binds to the local **ypserv**, which in turn satisfies all NIS requests originating on that machine.
- heterogenous environment
  - client hosts possess neither NIS maps nor ASCII files derived from them
  - look-ups are resolved across the network through NIS requests
  - the client's **ypbind** broadcasts a request for service and binds to the first responding **ypserv** process

## NIS utilities

---

A number of utilities are available to perform the following types of activities necessary for NIS administration:

- examine and manipulate NIS maps
- propagate NIS maps
- manage user accounts
- display information about the NIS domain and the servers it contains

A host must be bound in an NIS domain (that is, `ybind` must be running) before an NIS command can be executed from that host.

## NIS user accounts

---

Two types of accounts exist on an NIS network: local accounts and distributed accounts. Local accounts allow users access to the local host machine only, while distributed accounts allow users to access any machine on the NIS network. Each distributed account contains the same name and UID throughout the NIS network.

NIS password information is maintained by the following files in the `/etc` directory:

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>passwd.y</i>     | contains entries for distributed accounts. After entries are added using the <code>yppassmgmt</code> command, the <i>passwd</i> map is created from this file. Each NIS domain has a unique <i>passwd.y</i> file existing only on the master server.                                                                                                                                                                                                                                                                                                                                                                                                           |
| <i>passwd.local</i> | contains entries for local accounts. Entries are created by default when NIS is initialized on a system or added later using the <code>passmgmt</code> command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <i>passwd</i>       | contains entries for all active accounts, local and distributed. On the master server, <i>passwd</i> contains all entries from <i>passwd.y</i> and <i>passwd.local</i> . On slave and copy-only servers, it contains entries from <i>passwd.local</i> and ASCII translations of the <i>passwd</i> map received from the master.<br><br>Entries in <i>passwd.local</i> take precedence over entries in <i>passwd.y</i> . That is, if the two files include a user with the same name, the information in <i>passwd.local</i> is incorporated into <i>passwd</i> and the corresponding information in <i>passwd.y</i> (and the <i>passwd</i> map) is overridden. |

If you want to create local user accounts on an NIS server, do so before you configure NIS. Create these accounts using `sysadmsh(ADM)`. Refer to the *Administering Operating System Services* section of this book for more informa-

tion on `sysadmsh`. For information regarding account creation after initializing NIS, refer to the section “Managing user accounts” later in this chapter.

## *Understanding YP\_MAP\_X\_LATE*

---

The `/etc/yp/YP_MAP_X_LATE` file contains a complete listing of active NIS maps. The most common entries are *hosts*, *group*, *passwd*, and *ypservers*, although all standard and non-standard maps (that is, created by you) must be listed here if you wish NIS to serve them.

`ypinit(NADM)` consults `YP_MAP_X_LATE` when NIS is initialized and creates the maps listed therein. All desired maps must therefore be listed in `YP_MAP_X_LATE` before your system is initialized. If you add new maps after initialization, they need not be listed in `YP_MAP_X_LATE` to be made functional; however, because NIS consults `YP_MAP_X_LATE` when propagating maps, new maps must be listed here for them to be propagated using `yppush(NADM)`, `ypxfr(NADM)`, or other NIS utilities. For this reason, we recommend that you keep `YP_MAP_X_LATE` up-to-date on your master server.

Although slave and copy-only servers do not propagate maps and hence do not consult their `YP_MAP_X_LATE` files, we recommend that you keep these files current on non-master servers to simplify NIS administration.

The `YP_MAP_X_LATE` file also contains NIS aliases for NIS maps; aliases may be used by all NIS servers. If you have a system on your network that cannot handle long map names (that is, longer than 14 characters), you can add an alias to `YP_MAP_X_LATE`, showing a shortened alias for the map. If you have additional aliases to include in `YP_MAP_X_LATE`, add them before configuring NIS. The format of this file is described in the `ypmapxlate(NF)` manual page.

## *Security limitations on configuration*

---

To preserve the integrity of trusted databases, NIS may not modify certain sensitive files within a C2 Secure system. Although SCO NIS operates on a system with any of the security defaults (High, Improved, Traditional, and Low), the password and group databases are updated by NIS only in the security defaults that use `/etc/passwd` and `/etc/group` as the master databases. This condition is true if Traditional or Low security (Unsecure Mode) is chosen at installation time. If High or Improved security (Secure Mode) is chosen, `/tcb` manages `/etc/passwd` and `/etc/group` and does not allow NIS to update these files.

Only copy-only NIS servers may be initialized in Secure Mode settings on SCO systems. These servers may receive maps, but they may not propagate them. In addition, if a copy-only server in Secure Mode receives a password or group map, it cannot translate these maps into `/etc/passwd` or `/etc/group` ASCII



files. Table 10-1 lists the security configurations permitted for each type of SCO NIS server.

**Table 10-1 Security configurations for SCO NIS servers**

| SCO NIS server type | Permitted security settings                                             |
|---------------------|-------------------------------------------------------------------------|
| master              | Unsecure Mode only (Traditional or Low)                                 |
| slave               | Unsecure Mode only (Traditional or Low)                                 |
| copy only           | Secure Mode (High or Improved) or<br>Unsecure Mode (Traditional or Low) |

These security precautions help prevent unauthorized NIS password and group information from reaching SCO hosts. However, the same security precautions may not be available for non-SCO client implementations. Consult your documentation for non-SCO NIS client implementations if you are concerned with security.

**CAUTION** The installation of the NIS Runtime System compromises C2 security and reduces your ability to maintain a secure system. With NIS installed, it is relatively easy to change the master machine and change all the passwords on all systems in the network. Also, passwords (in encrypted form) are stored in text files and can be read by anyone with the correct privileges. If you are concerned with security, NIS may not be an appropriate service for your system.

## *Planning an NIS configuration*

When planning an NIS configuration, you must gauge your network administration needs with regard to domain structure, server disposition, security concerns, and the physical structure of the network. Your configuration will also be affected if the network is heterogenous, that is, if it includes non-SCO hosts. The following sections explain these considerations in more detail.

### *Selecting domains*

For most NIS installations, a single domain is sufficient to satisfy users' needs. However, if your site has multiple networks, you may wish to create separate domains for each network. A possible network division might be into accounting, marketing, and manufacturing domains.

In such an environment, it is recommended that there be one master and at least one slave server for each domain.

## Selecting servers

---

Two factors are important in selecting master and slave servers: uptime and synchronization. The master server should be a host that is stable and available for NIS requests. This suggests that a machine be dedicated as a master server and that there is at least one slave server for each master in a domain should the master become unavailable.

The master and slave servers should have CPU's comparable to those of the machines to which they will propagate maps. This is particularly important in heterogenous environments where a number of non-SCO NIS clients may seek to bind with the most available server. Because it is essential that current map information be uniformly available across the network, availability and equivalent speed of servers ensure that the propagation of map information is synchronized.

In an SCO NIS environment, slave servers can be used as a backup to the master server and to improve the ability to service map update requests from copy-only servers. In a heterogenous environment, slave servers should be added as needed to keep up with requests from non-SCO clients.

The physical structure of the network may influence the distribution of slave servers. For example, if you have a network that includes a bridge, you will probably want at least two slave servers on either side. This is because NIS requests are delayed when crossing bridges, such that a server on the same side of a bridge as the requesting host will satisfy a request sooner than one on the far side, even if the closer server is heavily loaded.

## Security

---

Although NIS is designed to administer password and group authorization files, such administration is incompatible with C2 security. SCO NIS maintains C2 security in an NIS environment by enforcing the following restrictions:

- only copy-only servers may be initialized on Secure Mode machines
- copy-only servers in Secure Mode may not use password and group maps for authentication. This means that master and slave servers are not afforded C2 protection.

In a heterogenous environment, SCO copy-only servers under C2 protection continue to refuse password and group maps from non-SCO master or slave servers. However, an SCO master or slave server (by definition in Unsecure Mode) may be able to propagate maps to C2-protected non-SCO clients. Consult your non-SCO client's documentation to address your security concerns. Refer also to the "Security under NIS" section later in this chapter.

# Configuring NIS

SCO NIS is enabled by setting up NIS servers, initializing NIS and starting the NIS daemons. The following sections explain these operations.

**NOTE** Please read this entire section before configuring NIS on your system.

## Setting up an NIS server

The following is an overview of the steps involved in setting up NIS servers:

1. Your network must be configured and operating properly before initializing NIS. Make sure the fully-qualified host name is set by running the `hostname` command. You should see the hostname and the default network domain name, for example, `jupiter.sco.com`. Usually, the network domain name and host name are set up by the system startup script.

**CAUTION** If your hostname and network domain name are not correct, you may have configuration problems with your network. However, changing the hostname and the default network name at this point may also force you to reconfigure other network services, such as `sendmail`, `MMDF` and `SCO LANManager`. Consult your *Administering Networking Services* guide for more information on network configuration.

2. Create local user accounts. If you do not create local user accounts before you run `ypinit` or `mkdev nis`, you must run `passmgmt` after initialization to make distributed accounts local.
3. Edit `YP_MAP_X_LATE` to remove names of files you do not wish NIS to service and add those you do. You should also modify NIS aliases for NIS maps as required. You must use NIS aliases if a map name exceeds the limit allowed by your operating system (usually 14 characters).
4. Ensure that the NIS domain name is set to `unknown` by entering `domainname` at your prompt. If the screen returns anything other than `unknown`, enter `domainname unknown`. `ypinit` and `mkdev nis` do not enter the appropriate command in the NFS startup script unless your domain name is set to `unknown`. `domainname(NC)` sets or displays the name of the current NIS domain.
5. Make sure the following files in `/etc` are complete and current on your master server: `passwd`, `hosts`, `ethers`, `group`, `networks`, `protocols`, and `services` (as well as `/usr/lib/mail/aliases` if you use `sendmail`). All files listed in `YP_MAP_X_LATE` should be available. If any of these files were removed from `YP_MAP_X_LATE`, do not be concerned.

6. Add the following entry to the **PATH** variable of your */profile* file (if you are running the Bourne or Korn shell) or */cshrc* file (if you are running the C shell):

```
/etc/yp
```

You are now ready to initialize your NIS server.

### Initializing NIS

---

The following steps explain how to initialize an NIS server. The procedure varies according to the type of server.

1. Log in as *root* and enter the following command to shut down the system:

```
/etc/shutdown -g0 -y
```

You see messages as the system services stop. Press **<Return>** at the **Safe to Power Off** message.

2. Type **y** and press **<Return>** when prompted to reset the location of the console. The following prompt appears:

```
Type <Ctrl>D to proceed with normal startup,
(or give root password for system maintenance)
```

3. Type the *root* password, then press **<Return>**.
4. Enter either of the following commands at your operating system prompt:

```
mkdev nis
```

or

```
/etc/yp/ypinit
```

The functionality and the options for **ypinit** and **mkdev nis** are identical.

5. NIS prompts you to choose the host type:

```
You must specify whether this host is to be a master, server,
or copy-only server. Please enter the server type:[mscq]
```

- Enter **m** to indicate that the local host is to be the NIS master.
- Enter **s** to indicate that you are setting up a slave database.
- Enter **c** to indicate that you are setting up a copy-only database.
- Enter **q** to discontinue initialization.

If NIS has been previously initialized on your system, you see the following:

```
NIS has been initialized previously on this machine
Please verify that you would like to re-initialize NIS
and then remove the file /etc/yp/ypinit_done
```

Remove the file `/etc/yp/ypinit_done`, reenter `mkdev nis` or `/etc/yp/ypinit`, and repeat this step.

If you attempt to configure a master or slave server on a host in Secure Mode (that is, at High or Improved security defaults), you see the following message:

```
ypinit: ERROR: Invalid configuration.
```

```
The security settings of this system are incompatible with the
NIS configuration requested. Please refer to your Release and
Installation Notes for further details. Initialization Aborted.
```

Refer to the `ypinit(NADM)` manual page for caveats concerning security restrictions.

#### 6. If you are initializing a master server, proceed to step 7.

If you are initializing a copy-only server in Secure Mode, you see:

```
Due to the restrictions of running NIS under the current security
setting of this system, this server will accept, but not integrate
sensitive maps received through NIS. Please refer to your Release
and Installation Notes for further details.
```

```
Hit return to continue
```

If you are initializing a slave or copy-only server, NIS prompts you to enter a host name. You see:

```
You must specify a host name to act as the NIS master
during map transfer [q to quit]:
```

Enter the name of a master server with an up-to-date and stable database.

#### 7. You then see:

```
The local host's domain name hasn't been set. Please set it.
Please enter the domain name or q to quit:
```

Enter the NIS domain name; you need an NIS domain name to retrieve data from an NIS database.

You see this prompt:

```
Installing the yp database will require that you answer a few
questions. Questions will all be asked at the beginning of
the procedure.
```

8. If this is the first installation and initialization of NIS or if you have previously removed NIS, proceed to step 9.

If an NIS database already exists on your machine, you see:

Can we destroy the existing domainname and its contents? [y/n/q: n]

Enter **y** to continue, which tells NIS to rebuild the previous NIS database with current information.

If you enter the default **n** or **q**, NIS initialization aborts, and you must manually remove the old NIS databases.

9. If you are initializing a copy-only server in Secure Mode, proceed to step 11.

NIS maintains local host accounts in */etc/passwd.local* and */etc/group.local*. For slave servers and copy-only servers, the host retrieves current NIS maps from the master server and merges the NIS maps with the local account information kept in */etc/passwd.local* and */etc/group.local*. The merged account information is stored in */etc/passwd* and */etc/group*. The network account information is then accessible on the local host. However, local account information overrides common entries, so NIS maps do not affect local maps.

If the */etc/passwd.local* file does not exist on the machine, you see the following message:

NIS stores local logins in */etc/passwd.local*, which currently doesn't exist on this machine. The */etc/passwd.local* file must be created before continuing.

Is it okay to copy */etc/passwd* to */etc/passwd.local*? [y/n/q: n]

Enter **y** to continue.

If you enter the default **n**, you must manually create the */etc/passwd.local* file, then start the initialization procedure again.

10. If the */etc/group.local* file does not exist on your machine, you see:

NIS stores local groups in */etc/group.local*, which currently doesn't exist on this machine. The */etc/group.local* file must be created before continuing.

Is it okay to copy */etc/group* to */etc/group.local*? [y/n/q: n]

Enter **y** to continue.

If you enter the default **n**, you must manually create the */etc/group.local* file, then start the initialization procedure again.

11. Next, verify that the `YP_MAP_X_LATE` file contains correct information.

You see:

```
The YP_MAP_X_LATE file contains the following:
map_names
```

```
Is this correct? [y/n/q: n]
```

If the list is not correct, enter the default `n`. Edit `YP_MAP_X_LATE` to contain the correct entries, and start the initialization procedure again. Refer to the `ypmapxlate(NF)` manual page for more information.

If the list is correct, enter `y` to continue.

12. If you are initializing a slave or copy-only server, proceed to step 14.

You see the following prompt:

```
The NIS network password file resides in /etc/passwd.yp on master
servers only. Since this node is being set up as a
master server, /etc/passwd.yp must be created.
```

```
Is it okay to create /etc/passwd.yp? [y/n/q: n]
```

Enter `y` to continue.

If you enter `n`, you must manually create the `/etc/passwd.yp` file, then start the initialization procedure again.

13. You see the following prompt:

```
The NIS network group file resides in /etc/group.yp
on master servers only. Since this node is being set up as a master
server, /etc/group.yp must be created.
```

```
Is it okay to copy /etc/group to /etc/group.yp? [y/n/q: n]
```

Enter `y` to continue.

If you enter `n`, you must manually create the `/etc/group.yp` file, then start the initialization procedure again.

## 14. You then see:

You must specify whether ypbind is to be run with the secure option (-s), the ypset option (-ypset), allowing anyone to change your binding, or the ypsetme option (-ypsetme) which only allows requests with *host-name's* IP address to change *host-name's* binding, or with no options.

Please see ypserv(NADM) for more information

- 1) ypbind -s
- 2) ypbind -ypset
- 3) ypbind -ypsetme
- 4) ypbind

Please enter your choice (1-4,q):

Enter the response appropriate to your security concerns. In an SCO NIS environment, the -ypsetme option (3) is recommended. In a heterogenous environment, the -s option (1) is appropriate in most cases. For further information, consult the ypserv(NADM) manual page, which includes information about ypbind.

## 15. If you are initializing a master server, proceed to step 16.

When the initialization procedure is complete, you see a series of messages resembling the following display when initializing a copy-only server:

There will be no further questions. The remainder of the procedure should take a few minutes to set up the NIS server.

The NIS maps will be transferred from *master\_server* when *server\_name* goes to run level 2.

If NIS has been previously initialized on your system, you see:

The backup directory /usr/lib/nisr already exists

Do you wish to continue with the SCO NIS Runtime System Initialization?

Enter y to continue; NIS then backs up system *passwd* files. If you enter n or q, NIS installation aborts.

You then see (for both slave and copy-only servers):

Backing up system passwd files ...

*server\_name's* NIS data base has been set up without any errors.

Your slave or copy-only server is now initialized. Proceed to step 18.



## 16: You then see:

At this point, we have to construct a list of the hosts which will be running NIS (NIS server hosts). The local host is in this list of NIS server hosts. Please continue to add the names for the other hosts, one per line. When you are done with the list, type a (Ctrl)D.

Enter all of the server names — master, slave, and copy-only — including all non-SCO server machines. The machine name and the Internet Protocol (IP) domain name are acceptable host names; the NIS domain name is not. Press (Return) after each host name. After entering the last host name, press (Ctrl)D to continue, and enter y to verify the host names are correct. NIS then creates and updates the maps.

When you initialize NIS, all of the UNIX system files that have the same names as NIS files are backed up in parallel subdirectories under */usr/lib/nisrt*. When you initialize the NIS server, NIS changes some system files. For example, NIS saves the system file */bin/passwd* to */usr/lib/nisrt/bin/passwd* and then replaces */bin/passwd* with a link to the file */etc/yppasswd*.

When the initialization procedure is complete, you see a series of messages resembling the following display:

There will be no further questions.  
The remainder of the procedure should take 5 to 10 minutes.

If NIS has been previously initialized on your system, you see:

The backup directory */usr/lib/nisrt* already exists

Do you wish to continue with the SCO NIS Runtime System  
Initialization?

Enter y to continue; NIS then backs up system *passwd* files. If you enter n or q, NIS installation aborts.

You then see:

```
Backing up system passwd files ...
```

```
Building /etc/yp/domain_name/ypservers...
```

```
Running /etc/yp/ypmake...
```

```
updated passwd
updated group
updated ethers
updated hosts
updated rpc
updated services
updated protocols
updated netgroup
updated aliases
updated ypbind
updated ypserv
updated yppasswd
```

*host\_name* has been set up as an NIS master server without any errors.

This display of updated maps should correspond to the maps listed in */etc/yp/YP\_MAP\_X\_LATE*.

17. The **ypinit** script then checks your tunable kernel parameters to see if adequate Streams buffers have been allocated for NIS. If not, it updates the Streams parameter and displays the following message:

```
Streams buffer sizes have been updated
```

```
Do you wish to relink the kernel now? (y/n/q)
```

Answer **y**. If you do not relink, the network may be slow or erratic because of insufficient resources. It takes several minutes to relink the kernel.

Your master server is now initialized.

18. Enter the following command to re-boot the system:

```
/etc/shutdown -g0 -y
```

Enter **(Ctrl)D** when prompted to return to multiuser mode.

The first time an NIS slave or copy-only server enters multiuser mode, its startup script calls **ypxfr** to transfer maps from the master server. Subsequent system startups do not transfer maps automatically.

**NOTE** If you wish to add distributed accounts, you must run **yppassmgt** on the master server. Refer to the section "Creating a distributed account" of this chapter for more information.

## Starting NIS

---

After you complete the installation and initialization procedures, you must activate NIS.

When you initialize NIS with `ypinit`, a copy of `/etc/nfs` is saved to `/usr/lib/nisr/etc/nfs` and the `/etc/nfs` script is modified to enable automatic startup and shutdown of NIS daemons. By default, installation of NFS places the startup script in `/etc/rc2.d/S89nfs` and the shutdown script in `/etc/rc0.d/K79nfs`. Both files are linked to `/etc/nfs`, such that NIS daemons are started and stopped whenever NFS is started or stopped.

After you have configured your machine as an NIS server and start NFS, it displays a series of messages similar to the following:

```
NFS Startup...
ONC daemons: portmap pcnfsd rwalld rusersd
NFS daemons: exportfs mountd nfsd(4) nfscld(4) biod(4) automount
Lock Manager: statd lckclnt(4) lockd
NIS services: ypserv ypbind yppasswd ypxfrd
NFS Startup complete.
The system is ready.
```

Other messages may also be displayed if you have configured your machine as a slave or copy-only server. For example, the following message indicates that the `protocols.byname` map was successfully converted from dbm to ASCII format:

```
Transferring proto.nam ...
map2ascii conversion for proto.nam map succeeded
```

Some map conversions, specifically for the `passwd` and `group` maps, will fail in Secure Mode.

If the startup of NFS does not display an NIS service message, either an error occurred during the initialization of NIS or `ypinit` has not modified `/etc/nfs` to enable the automatic starting of NIS daemons. Go back to single-user mode, determine the source of the problem, and run `ypinit` again.

## Managing NIS maps

---

The following sections explain how to modify existing NIS maps, propagate maps in an NIS domain, and create new NIS maps.

### Map manipulation utilities

---

SCO provides the following utilities to administer NIS maps:

- ypmake**      modifies the default maps listed in the `YP_MAP_X_LATE` file. **ypmake** also runs **yppush** by default to propagate newly updated maps.
- makedbm**    modifies only the *ypservers* map and any non-default maps created by the administrator. It can also be used to disassemble maps for examination in ASCII form.
- ypinit**       manipulates maps when NIS is being initialized but should not be used when NIS is already running

For more information, refer to the **ypmake(NADM)**, **makedbm(NADM)**, and **ypinit(NADM)** manual pages.

The SCO implementation of NIS provides **ypmake**, which searches for **make(CP)**. **make** is conventionally used to maintain groups of programs and is used by NIS to build and propagate maps. However, **make** is part of the SCO Development System and may not be installed on your system. If **make** is not available on your system, use **ypmake**, and modify the `/etc/yp/ypmake` file as you are instructed throughout this document.

If **make** is available on your system, you can issue either the **make** or **ypmake** command; however, **ypmake** finds and invokes **make** if available. **make** also requires a separate list of dependencies called the *Makefile*. Because this list is not included in **make** as it is in **ypmake**, you must edit the `/etc/yp/Makefile` file every time you are instructed to edit `/etc/yp/ypmake`.

**NOTE** We recommend that you use **ypmake** for NIS use even if **make** is already on your system. Because **ypmake** invokes **make**, you should keep the *ypmake* or *Makefile* files concurrent, particularly if both commands are available on your system.

## Modifying existing maps

To modify existing maps, you must change the databases that are served by NIS on the master server. Update the databases by first editing the ASCII file from which they are derived, then running **ypmake**. For example, you can modify the *hosts* map by following these steps:

1. Log into the master server.
2. Edit the */etc/hosts* file.
3. Issue the following commands to rebuild and propagate the map:

```
cd /etc/yp
./ypmake hosts
```

Use **ypmake** without arguments to modify and propagate all maps automatically.

**WARNING** Run **ypmake** or **make** only on master servers; unpredictable behavior in the NIS map set may result if they are run on slave or copy-only servers.

However, certain maps cannot be modified using **ypmake**. These are maps for which no ASCII file exists, such as the *ypservers* map. To modify these maps, use the **makedbm** command with the **-u** switch to unassemble them into a form that can be modified using standard tools (such as **awk**, **sed**, or **vi**). Build a new version using **makedbm** manually by either:

- redirecting the output of **makedbm** to a temporary file, modifying the file, and then using the file as input to **makedbm**. The following example could be used to add an entry to *ypservers*:

```
cd /etc/yp
makedbm -u ypservers > /tmp/ypservers
[Add an entry in /tmp/ypservers]
cat /tmp/ypservers | makedbm - /etc/yp/domain_name/ypservers
```

- piping the output of **makedbm** directly into **makedbm**. This is appropriate if the disassembled map can be updated by modifying it with, for example, **awk**, **sed**, or a **cat** append. The following example could be used to remove an entry from *ypservers*:

```
cd /etc/yp
makedbm -u ypservers | grep -v server_name \
| makedbm - /etc/yp/domain_name/ypservers
```

You must run the **yppush** command after modifying a map using **makedbm**; refer to the section “Adding servers” of this chapter for more information.

## Propagating an NIS map

---

Propagating a map means moving it from place to place, generally from the NIS master server to an NIS slave or copy-only server. Initially, the map is moved at system startup. After you initialize an NIS nonmaster server, `ypxfr(NADM)` transfers updated maps from the master server. The `ypxfr` command can be run in the following ways:

- periodically by `cron(C)`
- by `ypserv(NADM)`
- interactively by a user

In all cases, `ypxfr` checks for a file in `/etc/yp/map2ascii` with the same name as the transferred map. If this file exists, it is executed as the last step of the `ypxfr` process. These `map2ascii` files translate the new map into its ASCII counterpart and install it in its proper place. `map2ascii` routines are supplied for the most commonly used maps. You can write other routines and place them in the directory for use on other maps.

`ypxfr`'s transfer attempts and results can be captured in a log file. If `/usr/adm/nislog` exists, results are appended to it. The user is responsible for limiting the log file. To turn off logging, remove the log file. Refer to the `ypxfr(NADM)` manual page for more information about the log file.

Methods for transferring updated maps are described in the following sections.

### Using *cron*

Maps have differing rates of change; for instance, `protocols.byname` might not change for a long time, but `passwd.byname` might change several times a day in a large organization. You can set up `root`'s `crontab(C)` entries to run `ypxfr` periodically at a rate appropriate for any map in an NIS database. The `ypxfr` program contacts the master server and transfers the map only if the master's copy is more recent than the local copy.

To avoid a `crontab` entry for each map, group several maps into a shell script that can be run from a single `crontab` script file. Each shell script should contain maps with approximately the same characteristics. The following suggested groupings, mnemonically named, can be found in `/etc/yp`: `ypxfr1phr` (hourly), `ypxfr1pdy` (once daily), and `ypxfr2pd` (twice daily). If the rates of

change are inappropriate for a particular environment, you can easily modify or replace these shell scripts. The following **crontab** entries will call the scripts:

```
0 * * * * * /etc/yp/ypxfr1pdy
0 0 * * * * /etc/yp/ypxfr1pdy
0 0,12 * * * * /etc/yp/ypxfr2pdy
```

Run the same shell scripts at each NIS slave and copy-only server in the domain; do not run them from the master server. Alter the time of execution from one server to another to prevent the checking from impeding the master server. To transfer the map from a particular server, rather than the master server, use the **-h** option to **ypxfr** from within the shell script. You can check and transfer maps having unique change characteristics by explicit invocations of **ypxfr** within **crontab**.

### *Using the interactive approach*

To use the interactive approach, run **ypxfr** as a command. Typically, this is done only in exceptional situations: for example, when setting up a temporary NIS server to create a test environment, when transferring a new NIS map, or when quickly updating an NIS server that was out of service.

### *Using yppush on the master server*

The **ypxfr** command is also invoked by **ypserv** when the **yppush(NADM)** command is run on the NIS master server. **yppush** enumerates the NIS map **ypservers** to generate a list of NIS servers in the domain and sends a "Transfer Map" request to each named NIS server. The **ypserv** command spawns a copy of **ypxfr**, invoking it with the **-C** flag, and passes it the information needed to identify the map and call back the initiating **yppush** process with a summary status.

**NOTE** **yppush** requires that a version of a given NIS map already be present on the non-master systems before it can be transferred. Therefore, **yppush** cannot be used to propagate new maps; rather, **ypxfr** must be run on each non-master machine to request the new map.

## *Making new NIS maps*

---

This section explains how to create a sample non-standard NIS map. The sample map, which we will call *mymap*, consists of key-value pairs in which the keys are strings such as **key1**, **key2**, and **key3**, and the values are **value1**, **value2**, **value3**, and so on.

You can follow either of two procedures when creating new maps:

- Use an existing ASCII file as input.
- Use standard input.

After the map is created, you must also complete these steps:

1. Edit `YP_MAP_X_LATE` on all servers to include the new map.
2. Create a new `map2ascii` routine if you wish the new map to be converted to ASCII format after it is transferred.
3. Run `ypxfr` on each server to transfer the new map.

**CAUTION** Creating a new NIS map is an advanced network administration procedure. You should have a thorough understanding of NIS and be familiar with administrative tools such as `awk`, `sed`, and `grep` before proceeding.

### *Creating a new map from an ASCII file*

This section explains how to create a new NIS map from an ASCII file. An existing ASCII file named `/etc/yp/mymap.asc`, created with an editor or a shell script on the master server, is used as an example.

The ASCII file must be located in the subdirectory `home_domain` and contain a number of key-value pair strings. Create the NIS map for this file by entering the following commands:

```
cd /etc/yp
makedbm mymap.asc home_domain/mymap
```

If you want to add another pair (`key1,value1`) to the map, you can easily make the modification.

Change the map by first modifying the ASCII file. Modifications that you make to the map, rather than in the ASCII file, are lost. Make the modifications as follows:

```
cd /etc/yp
[make editorial change to mymap.asc]
makedbm mymap.asc home_domain/mymap
```

You may also wish to modify `/etc/yp/ypmake` or `/etc/yp/Makefile` on the master server so that the map can be conveniently rebuilt. Typically, an ASCII file is filtered through `awk`, `sed`, or `grep` to make it suitable for input to `makedbm`. Consult the existing `ypmake` file as a source for programming examples.

If `make(C)` is available on your system, use the mechanisms already in place in `/etc/yp/Makefile` when deciding how to create dependencies that `make` recognizes. Specifically, use `.time` files to see when the `Makefile` was last run for the map.



**NOTE** If you plan to use **ypmake** to maintain your new map, use the **NOPUSH** option the first time **ypmake** is run. Otherwise, **yppush** may hang when trying to access a server that does not have a copy of the new map.

## Creating a new map from standard input

If there is no original ASCII file, create the NIS map on the master server with the default domain *home\_domain* as shown:

```
cd /etc/yp
makedbm - home_domain/mymap
key1 value1
key2 value2
key3 value3
<Ctrl>D
```

To modify the map, use **makedbm** to create a temporary ASCII intermediate file as shown:

```
cd /etc/yp
makedbm -u home_domain/mymap > mymap.temp
```

At this point, edit *mymap.temp* so that it contains the correct information. Create a new version of the database with the commands:

```
makedbm mymap.temp home_domain/mymap
rm mymap.temp
```

## Propagating a new NIS map

To propagate a new NIS map, you must include copies of the map's **dbm** files in the domain directory on each NIS server in the domain. See the "Propagating an NIS map" section of this chapter for more information.

The */etc/yp/YP\_MAP\_X\_LATE* file contains a listing of all current maps in an NIS domain. Edit this file on all servers to include the new map before transferring the map.

If you translate the new map into an ASCII file and install it upon transfer, you need to write a **map2ascii** routine and place it in */etc/yp/map2ascii*. Consult an existing **map2ascii** routine as a source for programming examples.

To get an initial copy of the map, run the following command on each non-master server:

```
ypxfr -h new_master mymap
```

The **-h** forces **ypxfr** to get *mymap* from the master server. For subsequent transfers, **ypxfr** will read the map itself to find the master. Remember to add **crontab** entries if you want the new map transferred regularly.

**NOTE** The map must be globally available before non-SCO clients begin to access it. If the map is available from some NIS servers, but not all, you may observe unpredictable behavior from client programs.

## Managing user accounts

---

The following sections describe basic procedures for NIS user account administration.

### NIS account types

---

An NIS network has two kinds of user accounts on each computer: those that are *distributed* (and must be administered from the master machine) and those that are *local* (and must be administered from the local computer):

- distributed** provides account information that is uniform on each machine in an NIS domain. Changes made to distributed accounts are automatically propagated through NIS maps to other NIS nodes. You can easily manage distributed accounts from a central node, the master server.
- local** provides account information that can be unique to a particular machine. Because they are not affected by maps that are propagated between servers, local accounts allow you to exempt certain users from NIS management. In addition, changes that you make to local accounts do not affect NIS.

**NOTE** When NIS is installed, pre-existing accounts become local accounts by default, on both master and slave machines.

### Administration utilities

---

You need the following utilities to administer user accounts in an NIS network:

- sysadmsh** to create distributed accounts (on the master server only)
- passmgmt** to manage local accounts (on all hosts)
- yppassmgmt** to manage distributed accounts (on the master server only)

The **passmgmt** and **yppassmgmt** commands also modify, retire, lock, and unlock local information in the network password file. Refer to the *Administering Operating System Services* section of this book for more information on **sysadmsh**, and the **passmgmt(NADM)** and **yppassmgmt(NADM)** manual pages for more information on these commands.

**NOTE** You must be in multiuser mode (init state 2) with NIS daemons running before creating or modifying user accounts.

You should always run **ypmake** on the master server after making any change to user accounts.

## Creating NIS accounts

The following sections explain how to create distributed and local NIS accounts.

**WARNING** Do not make administrative accounts (for example, *root*, *MMDF*, or *UUCP*) distributed. Doing so may affect NIS functionality in unpredictable ways and seriously compromise the security of your network.

### Creating a distributed account

To create a new distributed account, follow these steps:

1. Log in to the master server machine.
2. Create a normal user account and home directory using **sysadmsh**.
3. This account is by default a distributed account. To propagate the new account to other NIS servers, enter:

```
cd /etc/yp
./ypmake passwd
./ypmake group
```

A distributed account requires only one home directory. However, a home directory must be available on all NIS-managed hosts through NFS or Automounter mounting. Refer to the corresponding chapters of this guide for more information.

**NOTE** A distributed account must be created on the master server only.

We recommend that you use **sysadmsh** to create distributed accounts under NIS. However, you can create distributed accounts directly using **yppassmgt** by following these steps:

1. Log in to the master server machine.
2. Determine the shell, account name, UID, GID, comment, and home directory. (Default UID and GID values will be assigned by **yppassmgt** if you do not include these options.) Refer to the **yppassmgt(NADM)** manual page for more information.

3. Make sure there are no conflicts with existing accounts, then enter the account information using the following command options:

```
yppassmgt -a -u uid -g gid -c comment \
-h homedir -s shell account_name
```

4. Assign an initial password using the `yppasswd` command.
5. Create a home directory using `sysadmsh`.
6. Run the following commands to distribute the account:

```
cd /etc/yp
./ypmake passwd
./ypmake group
```

### *Creating a local account*

To create a local account, follow these steps:

1. Log in to the machine where you want the local account.
2. Determine the shell, account name, UID, GID, comment, and home directory. (Default UID and GID values will be assigned by `passmgt` if you do not include these options.) Refer to the `passmgt(NADM)` manual page for more information.
3. Make sure there are no conflicts with existing accounts, then enter the account information using the following command options:

```
passmgt -a -u uid -g gid -c comment \
-h homedir -s shell account_name
```

4. Assign an initial password using the `passwd` command.
5. Create a home directory using `sysadmsh`.

## *Modifying NIS accounts*

---

The following sections explain how to modify NIS account status and password file entries.

### *Changing a distributed account to a local account*

To change a distributed account to a local account, use the `passmgt -n` option to add an entry to the `passwd.local` file on the local host. Enter:

```
passmgt -n account_name
```

Until you or the user change the password for the local account, the local account's attributes will still be the same as the distributed account's. This is because `passmgt` modifies the local `passwd.local` file only; it does not alter the distributed map.

If you wish to remove the user account from all machines except the local host, you must run `yppassmgt -d account-name` on the master server. However, it may be useful for the user to have both a distributed account and an account local to one machine, each with its own password.

### *Changing a local account to a distributed account*

To change a local account to a distributed account, you must first create a new distributed account on the master server, then remove the local account on the local machine. Follow these steps:

1. Log in to the master server.
2. Create the new account using `sysadmsh` and distribute it using `ypmake` (see the section “Creating a distributed account” earlier in this chapter).
3. Log in to the local host.
4. Run the following command to remove the account from `passwd.local`:  
`passmgt -d account_name`
5. When `ypmake` was run from the master server while there was still an existing local account, the entry in `passwd.local` filtered the incoming `passwd` map. This prevented the distributed account information from being incorporated into the local `/etc/passwd`. You must therefore run the following command to request a new version of the distributed account:

`/etc/yp/ypxfr passwd.nam`

### *Modifying password file entries*

Depending on the type of account, you can modify individual password file entries using the following command options:

- for distributed accounts (modifies `/etc/password.yp` on the master server):  
`yppassmgt -m options account_name`
- for local accounts (modifies `/etc/password.local` on the local server):  
`passmgt -m options account_name`

Except for the encrypted password field, you can modify all fields in the appropriate password files from the command line.

### *Removing password file entries*

You can remove individual password file entries using the following command options:

- for distributed accounts (removes entries from `/etc/password.yp` on the master server):  
`yppassmgt -d account_name`

- for local accounts (removes entries from */etc/password.local* on the local server):

```
passmgmt -d account_name
```

**NOTE** Do not use the commands *rmuser*, *rmgroup*, *rmpasswd*, or *unretire* in an NIS environment. Doing so may have unpredictable consequences for account administration.

## Administering an NIS network

---

The following sections discuss changing NIS server configuration troubleshooting procedures, and interoperability and security issues relevant to NIS network administration.

### Changing server configuration

---

The following sections explain how to add and remove servers and change the master server.

#### Adding NIS servers

After you initialize NIS, you can add servers to your NIS network at any time by:

1. editing the *ypservers* map on the master server and propagating it to other servers, and
2. installing and initializing NIS on the new server

Because the *ypservers* map is not built from an ASCII file, you must dump its contents and recreate it using NIS tools. The procedure is similar to that described in the “Modifying existing maps” section of this chapter.

The following steps explain how to add a new server:

1. Log into the master server in multiuser mode.
2. Enter the following command to dump the *ypservers* map into an ASCII file:

```
ypcat -k ypservers > /tmp/ypservers
```

3. Add the new server’s name to the list of servers in */tmp/ypservers*.
4. Change directories (*cd*) to the */etc/yp* and issue the following command to rebuild the *ypservers* map:

```
cat /tmp/ypservers | makedbm - /etc/yp/domainname/ypservers
```

5. Run the following command to force the propagation of the revised *ypservers* map:  
`yppush ypservers`
6. Log into the new server in system maintenance mode. If SCO NFS or the SCO NIS packages are not yet installed, you must first install them using *custom*; refer to the *Installation and Upgrade Guide* for more information.
7. Initialize NIS on the new server following the instructions in the “Configuring NIS” section of this chapter. When you bring the new server to multiuser mode, it automatically requests that all current maps be transferred.

## Removing servers

To remove a slave or copy-only server from an NIS domain you must:

1. remove the NIS package from the server using *custom*, and
2. edit the *ypservers* map on the master server and propagate it to other servers

**CAUTION** Do not use the following procedure to remove a master server. If you do, you must reconfigure your NIS network. Rather, refer to the “Changing a master server” section later in this chapter.

When SCO NIS is removed from a system, the */etc/passwd.local* file is removed. The */etc/passwd* file, which under NIS included both distributed and local accounts, becomes the authoritative password file. This means that distributed NIS accounts become regular user accounts when NIS is removed. If you want to restrict users with distributed NIS accounts from using the old server, you must remove their accounts before returning the old server to multiuser mode.

**NOTE** Although you can prevent NIS maps from reaching a server by killing its NIS daemon processes, the master still attempts to reach the server during map propagation, adding unnecessary burden on the system. For this reason, you should complete the following steps to remove NIS from a server.

Follow these steps to remove a server:

1. Log into the old server in system maintenance mode.
2. Run *custom* to remove the following NIS packages:
  - NISRT
  - NISMAN
3. If necessary, run *sysadmsh* to modify or retire previously distributed accounts.

4. If necessary, remove *root*'s **crontab** entries for NIS map transfers.
5. Reboot the system and return to multiuser mode. If kernel parameters were adjusted when NIS was removed, you are also prompted to relink the kernel.
6. Log into the master server in multiuser mode.
7. Edit the *ypservers* map as described in the section "Adding servers" to remove the old server's name.
8. Run the following command to force the propagation of the revised *ypservers* map:

```
yppush ypservers
```

### *Changing the master server*

You may need to change your master server, for example, if you rearrange server configuration or the master host is removed from service. Any server in the NIS domain can become the new master server, because each one has all active maps. However, because the old NIS master's name is contained in the existing map, you can neither use an existing copy at the new master nor send a copy there with **ypxfr**. Rather, you must recreate these maps by running **ypmake** (or **make**) on the new master server to associate the new server listing with each map.

The old master server must be running NIS to complete this procedure. This is because non-master servers consult their own copies of maps, which include the name of the old master server, when they receive a map transfer request originating from **yppush** on the master. New maps with the new master server listing must come from the old master server for the maps to be incorporated on slave and copy-only servers. Therefore, if the old master is not available when you change master servers, you must rebuild your entire NIS system from scratch.

**CAUTION** Changing master servers is a major reconfiguration of your network and should not be undertaken unless absolutely necessary. Make sure you understand the procedure thoroughly and back up your network configuration files before you begin.

The procedures described here assume that the new master is running SCO NIS. Other implementations of NIS configure servers in a different fashion; consult the documentation for your version of NIS before changing master servers.



To change NIS master servers, follow these steps:

1. Log into the new master server in multiuser mode.
2. Copy the */etc/passwd.yp* file from the old master. If this file is not available, you must recreate distributed accounts manually. If you have modified the */etc/yp/ypmake* or */etc/yp/Makefile* files, you must also copy them from the old master.
3. Run the following commands to rebuild the maps:

```
cd /etc/yp
rm *.time
ypmake -NOPUSH=1
```

By removing the timestamp files, you force the maps to be rebuilt regardless of prior modifications. The **NOPUSH** option to **ypmake** prevents maps from being pushed to other servers until you have completed this entire procedure.

4. If the map only exists as a **dbm** database, you can remake it on the new master by unassembling an existing copy from any NIS server and running the unassembled version back through **makedbm**. For example, enter:

```
cd /etc/yp
ypcat -k ypservers | makedbm - domain_name/ypservers
```

You must complete this step for the *ypservers* map and any other map you have created.

5. Log into the old master server.
6. Copy the new maps from the new master server with the following command:

```
/etc/yp/ypxfr -h new_master -f map_name
```

This command must be run on the old master for every current NIS map. The **-h** option forces **ypxfr** to transfer the map from the new master; the **-f** option forces the transfer even if the contents of the old and new maps are the same.

7. Propagate the new maps from the old master server with the following command:

```
/etc/yp/yppush map_name
```

This command must be run on the old master for every current NIS map. When this is completed, all non-master servers will have received maps including the new master listing.

8. If you are removing the old master server from NIS service, edit the *ypservers* map on the new master server following the instructions in the “Removing NIS servers” section of this chapter.

You can automate the map transfer procedure on the old master server using shell scripts and other tools. For example, the following shell script automatically completes steps 6 and 7:

```
:
MAPS='oawk '{print $1 ; }' /etc/yp/YP_MAP_X_LATE'
for map in $MAPS; do
 echo moving $map
 ypxfr -h new_master -f $map
 yppush $map
done
```

Make sure your `/etc/yp/YP_MAP_X_LATE` file is current before running this command; refer to the “Understanding `/etc/yp/YP_MAP_X_LATE`” section of this guide for more information.

## *Debugging NIS*

---

This section describes how to solve common NIS problems.

### *Checking TCP/IP*

Before troubleshooting what appears to be an NIS problem, verify that TCP/IP is up, configured, and running properly.

### *Checking the protected password file*

After you initialize and activate NIS at any time during system operation, you can detect errors and verify the integrity of the protected password database with the `/tcbin/authck` and the `/tcbin/integrity` commands. Refer to the `authck(ADM)` and the `integrity(ADM)` manual pages for more information.

### *Different versions of an NIS map*

Because NIS works by propagating maps among servers, different versions of a map may be found at servers on the network. This variation of map version occurs only during propagation and is normally transient.

Typically, normal update is prevented when an NIS server or gateway machine between NIS servers is down during a map transfer attempt. When all of the NIS servers and all of the gateways between them are running, `ypxfr` should succeed.

If a particular slave or copy-only server has problems updating, run `ypxfr` interactively on that server. If `ypxfr` fails, the screen displays a reason. This helps in understanding and fixing the problem. If `ypxfr` fails intermittently, create a log file to enable logging of messages.

The following program saves all output from `ypxfr`:

```
$ cd /usr/adm
$ touch nislog
```

The output resembles what `ypxfr` generates when it is run interactively, but each line in the log file is timestamped. The timestamp shows when `ypxfr` began its work. If copies of `ypxfr` ran simultaneously but their work took differing amounts of time, they can write their summary status line to the log files in an order different from that in which they were invoked. Any pattern of intermittent failure shows up in the log. When the problem is fixed, turn off logging by removing the log file. If you do not do this, the log file grows without limit.

On the misbehaving NIS slave or copy-only server, also inspect the `/usr/lib/crontab` file and the `ypxfr` shell scripts that it invokes. Typing mistakes in these files cause propagation problems, as do failures to refer to a shell script within `crontab`, or failures to refer to a map within any shell script.

Make sure the NIS slave or copy-only server is in the map `ybservers` within the domain. If not, the nonmaster server functions correctly as a server but is not notified by `yppush` when a new copy of a map exists.

### *ypserv crashes*

If the `ypserv` process crashes almost immediately and does not stay up even with repeated activations, the method for debugging is virtually identical to that described in the section “`ypbind` crashes” later in this chapter. Check the `portmap` daemon by entering:

```
$ ps -ef | grep portmap
```

If there is no `portmap`, reboot the local server by issuing the `shutdown` command. If the `portmap` daemon is present, enter the following:

```
$ rpcinfo -p
```

Output should be similar to the following example:

```
[program, version, protocol, port, service]:

[100000, 2, tcp, 111, portmapper]
[100000, 2, udp, 111, portmapper]
[100024, 1, udp, 1030, status]
[100024, 1, tcp, 1040, status]
[100020, 1, udp, 1038, llockmgr]
[100020, 1, tcp, 1041, llockmgr]
[100021, 2, tcp, 1042, nlockmgr]
[100021, 1, tcp, 1043, nlockmgr]
[100021, 1, udp, 1048, nlockmgr]
[100021, 3, tcp, 1044, nlockmgr]
[100021, 3, udp, 1053, nlockmgr]
```

The four entries representing the **ypserv** process are:

```
[100004, 1, udp, 1037, ypserv]
[100004, 1, tcp, 1046, ypserv]
[100004, 2, udp, 1051, ypserv]
[100004, 2, tcp, 1055, ypserv]
```

If they are not there, **ypserv** was unable to register its services. Reboot the machine. If they are there and change each time an attempt is made to restart */etc/ypserv*, reboot the machine.

## Commands hang

The most common problem at an NIS client node is that a command hangs and generates a console message:

```
NIS: server not responding for domain <odette>.
Still trying
```

Occasionally, commands hang even though the overall system appears to be working correctly and new commands can be run.

The preceding display indicates that **ypbind** on the local machine cannot communicate with **ypserv** in the domain *odette*. This often happens when machines that run **ypserv** crash. It can also occur if the network or the NIS server machine is so overloaded that **ypserv** cannot respond to the local machine's **ypbind** within the timeout period.

Under these circumstances, all of the other NIS client nodes on the net show the same or similar problems. The condition is temporary in most cases; the messages usually go away when the NIS server machine reboots and **ypserv** returns, or when the load on the NIS server nodes or the Ethernet decreases.

However, in the following circumstances, the condition does not change until the steps described are followed:

- The NIS client has not set, or has incorrectly set, **domainname** on the machine. Clients must use a domain name that NIS servers know. Use **domainname(NC)** to see the client domain name, and compare it with the domain name set on the NIS servers. The domain name should be set in */etc/rc2.d/S89nfs*. When */etc/rc2.d/S89nfs* fails to set or incorrectly sets **domainname**, enter the following commands as *root*:
  - Edit */etc/rc2.d/S89nfs* to correct the **domainname** line; the correct domain name enables the machine to boot properly.
  - Enter **domainname good\_domain\_name** to set the **domainname**.
- If the domain name is correct, check the local network to make sure that it has at least one NIS server machine. Machines can only bind to a **ypserv**

process on the local network, not on another accessible network. At least one NIS server for each machine's domain must be running on the local network. Having two or more NIS servers improves availability and response characteristics for NIS.

- If the local network has an NIS server, check it to make sure that it is running. Also check other machines on the local network. If several client machines have problems simultaneously, a server problem is likely. Try the `ypwhich` command on a client machine that is running; if `ypwhich` never returns an answer, kill the `ypwhich` process. To do so, find the process number using the `ps` command, then issue a kill command from another screen or terminal:

```
ps -ef | grep yp
kill -9 ypwhich_PID_#
```

(The second column of the `ps` display is the PID number.) The previous `ps` command also shows if there are `ypserv` and `ypbind` processes running. If the server's `ypbind` daemon is not running, start it by entering:

```
/etc/ypbind
```

If a `ypserv` process is running, enter a `ypwhich` on the NIS server machine. If `ypwhich` returns no answer, `ypserv` has probably hung and should be restarted. As super user, kill the existing `ypserv` process as described above and restart `ypserv` by issuing the command:

```
/etc/ypserv
```

If `ps` shows no `ypserv` process running, start one.

**NOTE** You must be logged in as *root* to start or kill `yp` processes. You can determine their status from multiuser mode using the `ps -ef` command, but only in Unsecure Mode.

## *NIS unavailable*

If NIS is unavailable on one machine even though other machines on the network appear to be functioning, many different symptoms may be displayed, including:

- Some commands appear to operate correctly, while others terminate and print an error message about the unavailability of NIS.
- Some commands enter a recovery mode particular to the program involved.
- Some commands or daemons crash with obscure messages or no message at all.

For example, messages such as the following might be displayed:

```
$ ypcat myfile
ypcat: can't bind to NIS server for domain <odette>.
Reason: can't communicate with ypbind.
```

```
$ /etc/yp/ypoll myfile
Sorry, I can't make use of NIS. I give up.
```

In such cases, run the following command on a directory containing files owned by many users (including users not in the local machine's */etc/passwd* file, for example */usr*):

```
$ls -l
```

If the *ls -l* command reports file owners not in the local machine's */etc/passwd* file as numbers, rather than names, this and similar symptoms usually indicate that NIS is not working and the *ypbind* process is not running. Use the command *ps -ef* to check for the process. If it is not found, enter the following command to start the process:

```
$ /etc/ypbind
```

NIS problems should disappear.

### ***ypbind crashes***

If *ypbind* crashes almost immediately each time it is started, problems probably lie in some other part of the system. Determine the presence of the *portmap* daemon by typing:

```
$ ps -ef | grep portmap
```

If it is not running, reboot the machine using */etc/shutdown*.

If *portmap* itself does not stay up or behaves strangely, problems are probably more fundamental. In this case, check the state of the network.

You may be able to talk to the *portmap* on the local machine from a machine operating normally. From such a machine, enter the following:

```
$ rpcinfo -p your_machine_name
```

If **portmap** is functioning correctly, the output should be similar to the following:

```
[program, version, protocol, port]:

[100005, 1, tcp, 1046 llockmgr]
[100001, 2, tcp, 1055 llockmgr]
[100001, 1, tcp, 1055 nlockmgr]
[100002, 1, tcp, 1052 nlockmgr]
[100008, 1, tcp, 1049 nlockmgr]
[100007, 1, tcp, 1027 nlockmgr]
[100007, 1, udp, 1026 nlockmgr]
[100007, 2, tcp, 1031 nlockmgr]
[100007, 2, udp, 1030 nlockmgr]
```

Note that port numbers are different on different machines. The four entries that represent the **ypbind** process are:

```
[100007, 1, tcp, port_# ypbind]
[100007, 1, udp, port_# ypbind]
[100007, 2, tcp, port_# ypbind]
[100007, 2, udp, port_# ypbind]
```

If these entries are not there, **ypbind** was unable to register its services. Reboot the machine. If the entries are there and change each time you attempt to restart */etc/ypbind*, reboot the system even if **portmap** is up.

## NIS interoperability

SCO systems resolve all lookup requests locally. The group and password entries "+" and "-" that allow and restrict network access are not supported by SCO systems. If your SCO system is acting as the master server, it should be able to resolve distributed lookup requests from non-SCO systems.

Regardless of whether your master server is on an SCO or a non-SCO system, all of the maps received by SCO systems are filtered before they are integrated (that is, translated into ASCII format). For example, */etc/yp/map2ascii/group.nam* filters incoming maps that are derived from */etc/group*. The filtering process eliminates:

- conflicts due to two or more users with the same UID
- entries found in *map.local* that supersede incoming map information
- the "+" entry that allows network access from incoming maps
- UIDs and GIDs outside the valid range. Look for the `MIN_ADMIN_UID` variable in the */etc/default/authsh* file for a list of valid ranges.

In addition, the */etc/yp/map2ascii/passwd.nam* and */etc/yp/map2ascii/group.nam* files check for security compatibility

**NOTE** If you wish to use NIS on a network that includes hosts configured with SCO UNIX System V C2 security specifications, verify that all of the maps provided by the master server are compatible with the more restrictive machines on the network. If the filtering process does not interpolate the maps compatibly with restrictive machines, you may have to edit the maps transferred by the non-SCO machine.

## *Security under NIS*

---

This section describes the files and entries involved in security issues.

### *Map integration restrictions in Secure Mode*

If your copy-only server is configured with one of the Secure Mode defaults, its *map2ascii* files will prevent any *passwd* or *group* maps from being integrated, regardless of their source.

**WARNING** Do not attempt to circumvent C2 security restrictions by removing security checks from the *map2ascii* files. Doing so may cause unpredictable behavior and is therefore an unsupported modification.

### *Special NIS password change*

When a password is changed with the *passwd(C)* command without NIS, the entry given in the system password database file is changed. To change a password in NIS, use the *yppasswd(NC)* command. To enable this service, you must start up the server daemon, *yppasswdd(NADM)*, on the machine serving as the master for the NIS password file. For convenience, the original */bin/passwd* file is moved to */usr/lib/nisrt/bin/passwd* and a link to */bin/yppasswd* is made. This means that when you invoke *passwd*, you are actually invoking *yppasswd*.

### *Manual pages covering security*

More detailed information on security issues can be found on the following manual pages:

- *passwd(C)*
- *export(NF)*
- *group(NF)*
- *netgroup(NF)*
- *yppasswd(NC)*
- *yppasswdd(NADM)*



## Directory of NIS services

---

The following list describes all of the NIS services and their corresponding manual pages.

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>domainname(NC)</b>  | sets or displays the name of the current NIS domain                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>makedbm (NADM)</b>  | a low-level tool for building an NIS map. It is normally called from <i>/etc/yp/ypmake</i> (or the <i>Makefile</i> when using <i>make</i> ), although it can also be used to modify maps for which no ASCII files exist or to create new maps. <i>makedbm</i> can also “disassemble” a map so you can see the key-value pairs it contains. The disassembled form can be modified with standard tools (such as editors, <i>awk</i> , <i>grep</i> , and <i>cat</i> ) and input back into <i>makedbm</i> .            |
| <b>map2ascii(NADM)</b> | describes the NIS commands that translate certain maps into their ASCII counterparts                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>passmgmt(NADM)</b>  | an NIS local account management utility that can create local accounts and change local accounts to distributed accounts                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>ypbind(NADM)</b>    | the NIS binder daemon. <i>ypbind</i> is a client agent that passes NIS requests to the server daemon. It must run on all machines that use NIS services, both servers and clients. Information on <i>ypbind</i> is included in the <i>ypserv(NADM)</i> manual page.                                                                                                                                                                                                                                                |
| <b>ypcat(NC)</b>       | dumps the contents of an NIS map. Use <i>ypcat</i> instead of <i>makedbm -u</i> in cases where it does not matter which server’s version is seen.                                                                                                                                                                                                                                                                                                                                                                  |
| <b>ypclnt(NS)</b>      | a group of function calls that provides an interface to NIS                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>ypfiles(NF)</b>     | describes the database structure of NIS                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>ypinit(NADM)</b>    | the NIS database initialization tool. Many maps must be constructed from files located in <i>/etc</i> such as <i>/etc/hosts</i> , <i>/etc/passwd</i> , and others. <i>ypinit</i> does all such construction automatically. In addition, it constructs initial versions of maps required by the system but not built from files in <i>/etc</i> , such as the map <i>ypservers</i> . Use this tool to set up all NIS servers for the first time. Do not use it as a general administrative tool for running systems. |

|                         |                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ypmake(NADM)</b>     | a script that builds several commonly changed components of the NIS database from ASCII files on the master and propagates maps to other servers. <b>ypmake</b> looks to see if <b>make(CP)</b> is available on your system; if it is available, <b>make</b> is called along with the <i>Makefile</i> script. If not, <b>ypmake</b> runs independently.                |
| <b>ypmapxlate(NF)</b>   | describes the <i>YP_MAP_X_LATE</i> file, which contains a complete listing of active NIS maps. It is used by NIS utilities when creating, modifying and propagating maps. It is also a translation table that handles NIS map names.                                                                                                                                   |
| <b>ypmatch(NC)</b>      | prints the value for one or more specified keys in an NIS map. There is no control over which server's version of the map is seen.                                                                                                                                                                                                                                     |
| <b>yppassmgmt(NADM)</b> | an NIS distributed account management utility. It can create and change distributed accounts, and change distributed accounts to local accounts.                                                                                                                                                                                                                       |
| <b>yppasswd(NC)</b>     | an NIS password administration utility that changes appropriate password databases. It uses the network to contact the master server for password changes on distributed accounts. Because <b>yppasswd</b> is linked to <b>passwd(C)</b> , NIS passwords are updated transparently using the conventional <b>passwd</b> command.                                       |
| <b>yppasswdd(NADM)</b>  | the daemon that handles password change requests from <b>yppasswd(NC)</b> to modify the NIS password file. This server is not run by default. If you want to enable remote password updating for the network information service, put an entry for <b>yppasswdd</b> in the <i>/etc/nfs</i> file on the master server. It cannot be run on slave and copy-only servers. |
| <b>yppoll(NADM)</b>     | asks any <b>ypserv</b> for information about a map that it holds internally                                                                                                                                                                                                                                                                                            |
| <b>yppush(NADM)</b>     | requests each of the <b>ypserv</b> processes within a domain to transfer a particular map, waits for a summary response from the transfer agent, then prints out the results for each server. Run <b>yppush</b> on the NIS master server only.                                                                                                                         |
| <b>ypserv(NADM)</b>     | the NIS database server daemon. <b>ypserv</b> satisfies Remote Procedure Calls (RPC's) for NIS services. It must run on each NIS server machine. The <b>ypserv(NADM)</b> manual page also contains information about the <b>ypbind</b> and <b>ypxfrd</b> daemons.                                                                                                      |

|                      |                                                                                                                                                                                                                                                                                                               |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ypset(NADM)</b>   | tells a <b>ypbind</b> process (by default, the local one) to get NIS services for a domain from a named NIS server                                                                                                                                                                                            |
| <b>ypwhich(NADM)</b> | tells you which NIS server a node is currently using for NIS, or which NIS server is master of a named map                                                                                                                                                                                                    |
| <b>ypxfr(NADM)</b>   | transfers an NIS map from the master to non-master servers, using NIS itself as the transport medium. You can run <b>ypxfr</b> interactively, or periodically from the <i>crontab</i> file. In addition, <b>ypserv</b> uses <b>ypxfr</b> as its transfer agent when propagating a map from the master server. |
| <b>ypxfrd(NADM)</b>  | the NIS map transfer daemon. <b>ypxfrd</b> satisfies requests from non-master servers to transfer maps. It must run on each NIS server machine. Information on <b>ypxfrd</b> is included in the <b>ypserv(NADM)</b> manual page.                                                                              |

## Chapter 11

# Automounter

---

Automounter allows remote NFS filesystems to be mounted automatically and transparently. This chapter explains how Automounter works and how to configure and administer Automounter in your network.

## Understanding Automounter

---

The chapter “NFS: Network File System” of this guide explains how to export and mount hierarchies through NFS with the **mount** command. You can also NFS mount file hierarchies using a different method — *automounting*.

The **automount** program enables users to mount and unmount remote directories on an as-needed basis. Whenever a user on a client machine running Automounter invokes a command that accesses a remote file or directory, such as opening a file with an editor, the hierarchy to which that file or directory belongs is mounted and remains mounted for as long as it is needed. No mounting is done at boot-time, reducing significantly the time needed to boot up. It is all done automatically and transparently.

## How Automounter works

---

Automounter actually mounts all file hierarchies under the directory */tmp\_mnt*. It then uses NFS’ symbolic link support to associate the actual mount point with the one in */tmp\_mnt*. This symbolic link is conceptually similar to a standard UNIX System V **link(ADM)** but it can be established between pathnames residing on different filesystems. The result is that the file hierarchies, actually mounted under */tmp\_mnt*, appear to be mounted on the correct mount points.

To illustrate a direct mount, let's assume that Automounter was configured to mount the remote file hierarchy *bigmo:/usr/man* on the local directory */usr/man*. The first user to issue the command `cd /usr/man` causes Automounter to perform the equivalent of the following commands:

1. `mkdir /tmp_mnt/usr/man`

If the */tmp\_mnt* or */tmp\_mnt/usr* directories do not exist at the time, Automounter creates them as well.

2. `mount -f NFS bigmo:/usr/man /tmp_mnt/usr/man`

Automounter performs a conventional NFS mount.

3. `ln -s /tmp_mnt/usr/man /usr/man`

The `ln -s` command shows a symbolic link operation associating */usr/man* and */tmp\_mnt/usr/man*. However, Automounter does not really create symbolic links. Rather, it acts as a symbolic link server, in this case redirecting access requests destined for */usr/man* to */tmp\_mnt/usr/man*.

As a result, *bigmo:/usr/man* appears to be directly NFS-mounted on the local */usr/man* directory.

Figure 11-1 illustrates direct mounting, showing how Automounter can mount automatically the online UNIX manual pages from a specified server.

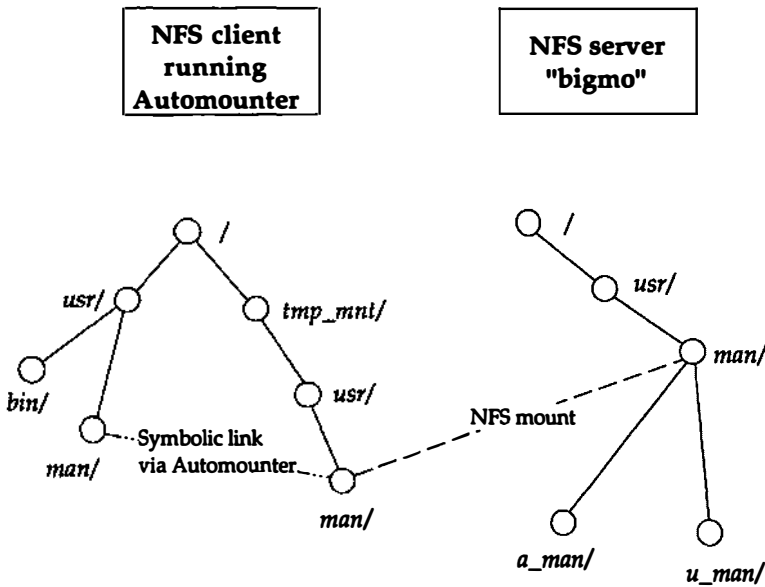


Figure 11-1 An example of direct mounting using Automounter

To illustrate an indirect mount, let's assume that Automounter is configured such that the remote hierarchies *lucy:/doc.stuff/spellers* and *alice:/doc/bin/roff* appear to be mounted on the local directory */doc\_tools*. When a user issues the commands */doc\_tools/spellers/mspell* and */doc\_tools/roff/nroff*, it causes Automounter to perform the equivalent of the following commands:

1. **mkdir /tmp\_mnt/doc\_tools/spellers**  
**mkdir /tmp\_mnt/doc\_tools/roff**

If the */tmp\_mnt* or */tmp\_mnt/doc\_tools* directories do not exist at the time, Automounter creates them as well.

2. **mount -f NFS lucy:/doc.stuff/spellers /tmp\_mnt/doc\_tools/spellers**  
**mount -f NFS alice:/doc/bin/roff /tmp\_mnt/doc\_tools/roff**

Automounter performs conventional NFS mounts.

3. **ln -s /tmp\_mnt/doc\_tools/spellers /doc\_tools/spellers**  
**ln -s /tmp\_mnt/doc\_tools/roff /doc\_tools/roff**

Again, Automounter does not really create symbolic links. Rather, it acts as a symbolic link server, in this case redirecting access requests destined for */doc\_tools* to */tmp\_mnt/doc\_tools*.

**NOTE** The directory */doc\_tools* in this example is created and managed by Automounter; it does not exist on any disk.

Figure 11-2 illustrates indirect mounting, showing how Automounter can assemble and automatically mount a directory of tools from specified servers.

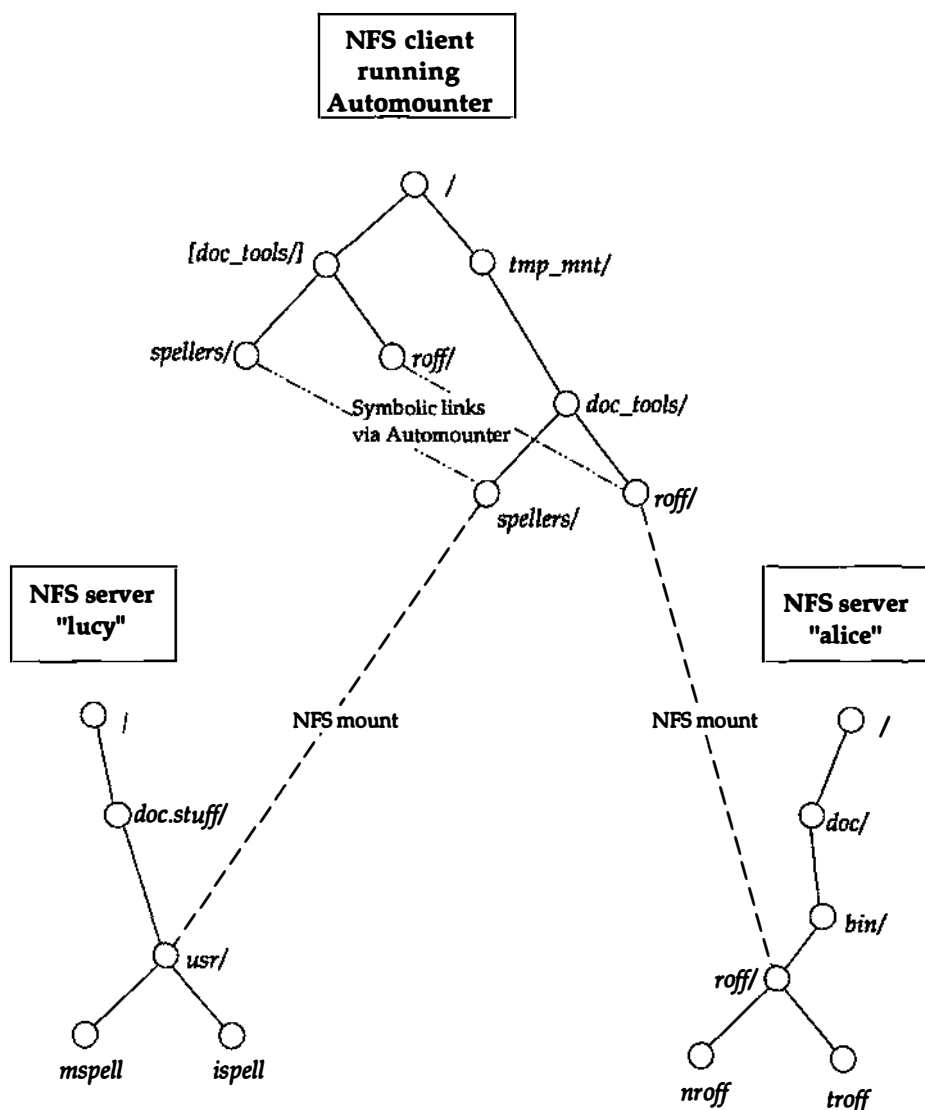


Figure 11-2 An example of indirect mounting using Automounter

A comparison of these examples illustrates the major differences between direct and indirect mounting:

- direct mounting
  - mounts complete directories
  - is symbolically linked to a subdirectory in a local filesystem without hiding other existing subdirectories
- indirect mounting
  - assembles discontinuous subdirectories under a virtual parent directory
  - creates a parent directory that cannot contain files other than those managed by Automounter

Thus, in our examples, a directly automounted `/usr/man` directory can coexist with a local `/usr/lib` directory. However, a local `/doc_tools` directory (if it existed previous to Automounter configuration) is hidden by an indirectly automounted `/doc_tools/spellers` directory.

## Automounter maps

Automounter determines which mount points to monitor and file hierarchies to mount from a special set of files called *maps*. Maps are configuration files that describe the possible configurations for NFS mounts on the system. These maps can reside on the local machine or be managed via NIS.

There are three types of Automounter maps:

|          |                                                                                                                                                                                                                                                                             |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| direct   | lists complete directories and their locations. Whenever any node in the directory hierarchy is accessed, Automounter mounts the entire directory.                                                                                                                          |
| indirect | lists individual directory entries and their locations. Automounter creates a virtual parent directory from subdirectories that may reside on different devices or hosts. Whenever any node in the subdirectory hierarchy is accessed, Automounter mounts the subdirectory. |
| master   | lists all indirect and direct maps, applicable options and mount points. If a master map <i>auto.master</i> is present when the system boots up, Automounter reads the master map and starts a daemon for the specified mount point.                                        |

Direct automounting is similar to conventional NFS mounting in that an entire filesystem is exported. For example, if the filesystem `/usr/man` on the server *bigmo* is listed in a direct map on the client, it is mounted automatically whenever a user on the client accesses the filesystem. This may be done, for instance, by issuing the command `man rpcinfo`, which accesses the file `./cat.NADM/rpcinfo.NADM` in the `/usr/man` directory. The mount point in this



example appears to the user to be */usr/man*, and as in any other mount procedure, the existing directory entries are hidden by the directory mounted on top of it.

Indirect mounting allows you to distribute resources to different devices or hosts while maintaining the appearance to the user that these resources belong to the same filesystem. For example, a user may want access to a number of documentation tools, including layout tools, fonts, and spell checkers. The system administrator can use Automounter to make it appear that all these resources are available in the */doc\_tools* directory of the user's home machine, even though each resource could be on a different device or host.

When **automount** is started, either from the command line or */etc/nfs*, it forks a daemon to serve the mount points specified in the maps. It does this by establishing itself as the NFS server for the specified mount points. When one of these mount points is accessed or crossed, Automounter fields the NFS protocol request as would the real NFS server daemon, **nfsd**(NADM). It then mounts the appropriate remote hierarchy, as specified in Automounter maps.

When a predetermined amount of time has elapsed without the hierarchy being accessed, Automounter automatically unmounts it.

### *Planning an Automounter configuration*

---

A typical use for Automounter is an installation of several large server systems and 10 or more workstations where NIS is being used. NFS mounts for each user on each workstation are managed centrally from one of the servers using NIS, which also allows distributed access to a shared database of user accounts. Automounter ensures that users can access essential system files regardless of the workstation from which they log in to the system. In a larger system, you might use NIS to accomplish such transparency without additional administration on each workstation or the performance burden represented by dozens of active NFS mounts.

Automounter is also useful on a smaller NFS network where hosts must double as clients and servers. NIS management might not be necessary in such a configuration, but Automounter can still reduce the number of active mounts and improve network access and efficiency.

Automounter also allows a client to configure redundant servers. That is, it allows a client to configure a set of servers from which a particular file hierarchy can be mounted. Therefore, in the direct mount example (Figure 4-1), instead of just specifying *bigmo* as the remote server for */usr/man*, it is possible to specify remote servers *bigjoe* and *bigben*, as well. If configured in this fashion, Automounter locates the first available server (using an RPC query), then mounts */usr/man* from that system.

## Configuring Automounter

---

To configure Automounter into an NFS network, you must:

- ensure that the remote directories you wish to mount are listed in the */etc/exports* files on your servers (see the “NFS: Network File System” chapter of this guide for more information)
- create at least one Automounter map in the */etc* directory on the client
- start Automounter by issuing the **automount** command, either on the command line or through an entry in the */etc/nfs* start-up script

The following sections explain how to prepare Automounter maps and start Automounter.

**NOTE** We recommend that you use indirect maps whenever possible, because they can be modified more easily and they create fewer entries in the mount table (*/etc/mnttab*).

We also recommend that you create a master map. Doing so simplifies Automounter administration and enables Automounter to start automatically when your system goes to multiuser mode.

### Preparing the maps

---

Unlike the **mount** command, **automount** does not consult the */etc/default/filesys* file for information on which file hierarchies to mount. Rather, it consults the map file(s) specified on the command line at startup time. If no maps are specified, it looks for an NIS map called *auto.master*. If the NIS *auto.master* is not present, **automount** exits silently.

By convention, all Automounter maps are located in the */etc* directory and have filenames prefixed with *auto*.

#### Writing a master map

Each line in the master map, by convention called */etc/auto.master*, has the syntax:

**mount-point**    **map-name**    [**mount-options**]

Here:

**mount-point** is the full pathname of a directory. If the directory does not exist, Automounter creates it if possible. If the directory exists and is not empty, mounting on it hides its contents. Automounter issues a warning message in this case.

- map-name** is the map Automounter uses to find the mount points and locations.
- mount-options** is an optional comma-separated list of options that regulate the mounting of the entries mentioned in *map-name*, unless the entries in *map-name* list other options

A line whose first character is “#” is treated as a comment, and everything that follows until the end of the line is ignored. A backslash “\” at the end of the line permits splitting long lines into shorter ones. The notation “/-” as a mount point indicates that the map in question is a direct map and no particular mount point is associated with the map as a whole.

A typical *auto.master* file contains:

```

/- /etc/auto.direct -ro
/home /etc/auto.home -rw
/net -hosts
```

Automounter recognizes the following special mount points and maps:

#### Mount point “/-”

In the earlier example, the mount point /- is a filler that Automounter recognizes as a directive not to associate the entries in */etc/auto.direct* with any directory. Rather, the mount points are the ones mentioned in the map. (Remember, in a direct map the key is a full pathname.)

#### Mount point /home

The mount point */home* is the directory under which the entries listed in */etc/auto.home* (an indirect map) are to be mounted. That is, they are mounted under */tmp\_mnt/home*, and the NFS symbolic link machinery is used to associate */home/directory* and */tmp\_mnt/home/directory*.

*/home* is the preferred convention for automounting home directories. Doing so allows you to provide users with their familiar home environment regardless of the the host from which they log in to the system.

#### Mount point /net

Automounter mounts under the directory */net* all the entries under the special map, *-hosts*. This is a built-in map that does not use any external files except the hosts database */etc/hosts* or the NIS map, *hosts.byname*, if NIS is running. The */net* convention affords the user transparent access to any NFS server on the network (where the client’s access permissions match those listed in */etc/exports*).

**NOTE** Because Automounter does not mount the entries until needed, the specific order is not important. Once the automount daemon is in place, a user entering the following command changes directory to the top of the file hierarchy (that is, the *root* filesystem) on the machine *gumbo*,

```
cd /net/gumbo
```

This is true as long as the machine is in the hosts database and it exports any of its filesystems. However, the user may not see all the files and directories under */net/gumbo*. This is because Automounter can mount only the exported filesystems of host *gumbo*, in accordance with the restrictions placed on the exporting. in */etc/exports*.

Automounter performs the following tasks in response to the previous command.

1. Query the server's mount service using the RPC null procedure to see if it is responding.
2. Request the list of exported hierarchies from the server.
3. Sort the exported list according to length of the pathname.

```
/usr/src
/export/home
/usr/src/scs
/export/root/blah
```

This sorting ensures that the mounting is done in the proper order, that is, */usr/src* is done before */usr/src/scs*.

4. Proceed down the list, mounting all the filesystems at mount points in */tmp\_mnt* (creating the mount points as needed).

**NOTE** Automounter has to mount all of the filesystems that the server in question exports. Therefore, if the user issues the following *ls* command:

```
ls /net/gumbo/usr/include
```

Automounter mounts all of *gumbo's* exported systems, not just */usr*.

In addition, unmounting that occurs after a certain default period is from the bottom up. This means if one of the directories at the top is busy, Automounter has to remount the hierarchy and try again later.

The *-net* special map thus provides a very convenient way for users to access directories in many different hosts without having to use *rlogin* or *rsh*. Furthermore, they no longer have to modify their */etc/default/filesys* files or mount the directories by hand as superuser.

Notice that both */net* and */home* are arbitrary names dictated by convention. Automounter creates them if they do not exist already.

## Writing an indirect map

The syntax for an indirect map is:

**key**      [**mount-options**]      **location**

Here:

**key**                      is the pathname of the mount point.

**mount-options**      are the options you want to apply to this particular mount.

**location**              is the location of the resource, specified as **server:pathname[:subdirectory]**.

**key** is the basename (not the full pathname) of the directory that is used as mount point. Once the key is obtained by Automounter, it is suffixed to the mount point associated with it either by the command line or by the master map that invokes the indirect map in question.

As in the master map, a line whose first character is “#” is treated as a comment and everything that follows until the end of the line is ignored. A backslash at the end of line permits splitting long lines into shorter ones.

The only formal difference between a direct and an indirect map is that the key in a direct map is a full pathname, whereas in an indirect pathname it is a simple name (no slashes). For instance, the following is an entry in a direct map:

```
/usr/man -ro goofy : /usr/man
```

The following is an entry in an indirect map:

```
parsley -ro veggies : /usr/greens
```

Clearly, the **key** in the indirect map requires more information, specifically the actual location of the mount point, *parsley*. This information must be provided at the command line or through another map. For instance, if the above line is part of a map called */etc/auto.veggies*, the appropriate level of qualification can be provided through invoking **automount** as follows:

```
automount /veggies /etc/auto.veggies
```

The same qualification can be achieved by adding the following specification to the master map:

```
/veggies /etc/auto.veggies -ro, soft, nosuid
```

In either case, the mount directory */veggies* qualifies the entries in the indirect map */etc/auto.veggies*. The end result is that the hierarchy */usr/greens* from the machine *veggies* is mounted on */veggies/parsley*, when needed.

For instance, one of the entries in the master map presented above as an example, reads:

```
/home /etc/auto.home -rw
```

Here */etc/auto.home* is the name of the indirect map that contains the entries to be mounted under */home*.

A typical *auto.home* map might contain:

```

willow willow : /home/willow
cypress cypress : /home/cypress
poplar poplar : /home/poplar
pine pine : /export/pine
apple apple : /export/home
ivy ivy : /home/ivy
peach peach : /export/home
 -rw, nosuid

```

As an example, assume that the map above is on the host *oak*. If the user, *laura*, has an entry in the password database specifying her home directory as */home/willow/laura*, then whenever she logs into machine *oak*, Automounter mounts (as */tmp\_mnt/home/willow*) the directory */home/willow* residing in machine *willow*. If one of the directories is indeed *laura*, she is in her home directory, which is mounted read/write, as specified by the options field in the master map entry.

Suppose, however, that *laura*'s home directory is specified as */home/peach/laura*. Whenever she logs into *oak*, Automounter mounts the directory */export/home* from *peach* under */tmp\_mnt/home/peach*. Her home directory is mounted read/write, *nosuid*. Any option in the file entry overrides all options in the master map or the command line.

Now, assume the following conditions occur:

- User *laura*'s home directory is listed in the password database as */home/willow/laura*.
- The machine *willow* exports its *home* hierarchy to the machines mentioned in *auto.home*.
- All those machines have a copy of the same *auto.home* and the same password database.

Under these conditions, the user *laura* can run **login** or **rlogin** on any of these machines and have her home directory mounted in place for her.

Furthermore, now *laura* can also enter the command:

```
cd ~bigmo
```

Automounter mounts *bigmo*'s home directory for her (if all permissions apply).

On a network without NIS, you have to change all the relevant databases (such as */etc/passwd*) on all systems on the network to accomplish this. On a network running NIS, make the changes on the NIS master server and propagate the relevant databases to the slave servers.

## Writing a direct map

The syntax for a direct map (like that for an indirect map) is:

**key**      [**mount-options**]      **location**

Here:

**key** is the full pathname of the mount point. (Remember that in an indirect map, this is not a full pathname.)

**mount-options** are optional but, if present, override the options of the calling line or the defaults for the entry in question. (See the section "Starting Automounter" later in this chapter.)

**location** is the location of the resource, specified as **server:pathname[:subdirectory]**.

The following is a sample */etc/auto.direct* map:

```
/usr/local \
 /bin -ro, soft ivy : /usr/local/bin \
 /src -ro, soft ivy : /usr/local/src
/usr/man -ro, soft oak : /usr/man \
 rose : /usr/man \
 willow : /usr/man
/usr/games -ro, soft peach : /usr/games
/usr/spool/news -ro, soft pine : /usr/spool/news
```

Two unusual features in this map, multiple mounts and multiple locations, are discussed in the next two subsections.

### Multiple mounts

A map entry can describe a multiplicity of mounts, where the mounts can be from different locations and with different mount options. Consider the first entry in the previous example:

```
/usr/local \
 /bin -ro, soft ivy : /usr/local/bin \
 /src -ro, soft ivy : /usr/local/src
```

This is, in fact, one long entry whose readability has been improved by splitting it into three lines by using the backslash and indenting the continuation lines with blank spaces or tabs. This entry mounts */usr/local/bin* and */usr/local/src* from the server *ivy*, with the options read-only and soft. The entry could also read:

```
/usr/local \
 /bin -ro, soft ivy : /usr/local/bin \
 /src -ro oak : /usr/local/src
```

Here, the options are different and more than one server is used.

The preceding example might also be represented with two separate entries, for example:

```
/usr/local/bin -ro, soft ivy : /usr/local/bin
/usr/local/src -ro oak : /usr/local/src
```

The difference is that the first case, a multiple mount, guarantees that both directories will be mounted when you reference one of them. In the case of the separate entries, if you reference one directory you cannot change (cd) to the other using a relative path, because it is not mounted yet.

```
cd /usr/local/bin
cd ../src
../src: No such file or directory
```

A multiple mount obviates this problem. In multiple mounts, each file hierarchy is mounted on a subdirectory within another file hierarchy. When the root of the hierarchy is referenced, Automounter mounts the whole hierarchy.

A special case of multiple mounts is when the root of a hierarchy has to be mounted as well. This is called a "hierarchical" mount. The following illustration shows a true hierarchical mounting:

```
/usr/local \
/ -rw peach : /usr/local \
/bin -ro, soft ivy : /usr/local/bin \
/src -ro oak : /usr/local/src
```

**NOTE** A true hierarchical mount can be problematic if the server for the root of the hierarchy goes down. Any attempt to unmount the lower branches fails, because the unmounting has to proceed through the mount root, which also cannot be unmounted while its server is down.

If multiple mounts share the same options, it is not necessary to duplicate these options for each server. Rather, the options precede the different mounts, as follows:

```
/usr/local -ro, soft \
/bin ivy : /usr/local/bin \
/src oak : /usr/local/src
```

If one of the mount points needs different specification, modify only its options:

```
/usr/local -ro, soft \
/bin ivy : /usr/local/bin \
/src -rw oak : /usr/local/src
```



### Multiple locations

The mount point */usr/man* in the following direct map lists more than one location.

```

/usr/local \
 /bin -ro , soft ivy : /usr/local/bin \
 /src -ro , soft ivy : /usr/local/src
/usr/man -ro , soft oak : /usr/man \
 rose : /usr/man \
 willow : /usr/man
/usr/games -ro , soft peach : /usr/games
/usr/spool/news -ro , soft pine : /usr/spool/news

```

This means that the mounting can be done from any of the replicated locations, in this case, *oak*, *rose*, or *willow*. This list can also be expressed as a comma-separated list of servers, followed by the colon and the pathname (as long as the pathname is the same for all of the replicated servers), for example:

```

/usr/man -ro , soft oak, rose, willow : /usr/man

```

The first server to respond to the RPC ping issued by Automounter is selected, and an attempt is made to mount from it. The list does not imply an ordering, although servers on the local network will be contacted first.

This redundancy, very useful in an environment where individual servers may or may not be exporting their file systems, exists only at mount time. There is no status checking of the mounted-from server by Automounter once the mount occurs. If the server goes down while the mount is in effect, the filesystem becomes unavailable. An option here is to wait five minutes until the auto-unmount takes place and try again. Next time around, Automounter chooses one of the other, available servers. Another option is to use the **umount** command, inform Automounter of the change in the mount table (as specified in the later section, “Refreshing the mount table,” and retry the mount.

**CAUTION** Care should be taken when using multiple locations for read/write file hierarchies, because the actual server may change from access to access.

## Specifying subdirectories

The earlier subsection “Writing an indirect map” showed the following typical *auto.home* file:

```
willow willow : /home/willow
cypress cypress : /home/cypress
poplar poplar : /home/poplar
pine pine : /export/pine
apple apple : /export/home
ivy ivy : /home/ivy
peach -rw, nosuid peach : /export/home
```

Given this *auto.home* indirect file, every time a user wants to access a home directory in, for example, */home/willow*, all the directories under it are mounted. Another way to organize an *auto.home* file is by user name:

```
john willow:/home/willow/john
mary willow:/home/willow/mary
joe willow:/home/willow/joe
```

The above example assumes that home directories are of the form */home/user* rather than */home/server/user*. If a user now enters the following command:

```
ls ~john ~mary
```

Automounter has to perform the equivalent of the following actions:

```
mkdir /tmp_mnt/home/john
mount -f NFS willow:/home/willow/john /tmp_mnt/home/john
ln -s /tmp_mnt/home/john /home/john

mkdir /tmp_mnt/home/mary
mount -f NFS willow:/home/willow/mary /tmp_mnt/home/mary
ln -s /tmp_mnt/home/mary /home/mary
```

It is possible to optimize the work done by Automounter by modifying the entries in the *auto.home* map to use the optional “subdirectory” field of the *location*. The new *auto.home* map looks like:

```
john willow : /home/willow : john
mary willow : /home/willow : mary
joe willow : /home/willow : joe
```

Here “john,” “mary,” and “joe” are entries in the “subdirectory” field. Now, when a user refers to *john*’s home directory, Automounter mounts *willow:/home/willow* and links */tmp\_mnt/home/willow/john* and */home/john*.

If the user then requests access to *mary*’s home directory, Automounter sees that *willow:/home/willow* is already mounted, and simply links

`/tmp_mnt/home/willow/mary` and `/home/mary`. In other words, Automounter now only performs the equivalent of the following:

```
mkdir /tmp_mnt/home/john
mount -f NFS willow:/home/willow /tmp_mnt/home
ln -s /tmp_mnt/home/john /home/john
ln -s /tmp_mnt/home/mary /home/mary
```

In general, it is a good idea to provide a *subdirectory* entry in the *location* when different map entries refer to the same mounted filesystem from the same server.

## Metacharacters

Automounter recognizes some characters as having a special meaning. Some are used for substitutions, some to escape other characters.

### Ampersand "&"

Suppose you have a map with a lot of subdirectories specified, for example:

```
john willow:/home/willow:john
mary willow:/home/willow:mary
joe willow:/home/willow:joe
able pine:/export/home:able
baker peach:/export/home:baker
[. . .]
```

Consider using string substitutions. You can use the ampersand character "&" to substitute the key wherever it appears.

Using the ampersand, the above map now looks as follows:

```
john willow:/home/willow:&
mary willow:/home/willow:&
joe willow:/home/willow:&
able pine:/export/home:&
baker peach:/export/home:&
[. . .]
```

Suppose the name of the server is the same as the key itself, for instance:

```
willow willow:/home/willow
peach peach:/home/peach
pine pine:/home/pine
oak oak:/home/oak
poplar poplar:/home/poplar
[. . .]
```

The use of the ampersand results in:

```
willow & : /home/&
peach & : /home/&
pine & : /home/&
oak & : /home/&
poplar & : /home/&
 [. . .]
```

### Asterisk "\*\*\*

Notice that all the above entries have the same format. This permits you to use the catch-all substitute character, the asterisk ("\*"). The asterisk reduces the whole thing to:

```
* & : /home/&
```

Here each ampersand is substituted by the value of any given key. Notice that once Automounter reads the catch-all key it does not continue reading the map, so that the following map is viable:

```
oak & : /export/&
poplar & : /export/&
* & : /home/&
```

In the next map the last two entries are always ignored:

```
* & : /home/&
oak & : /export/&
poplar & : /export/&
```

You could also use key substitutions in a direct map, in situations like the following:

```
/usr/man willow, cedar, poplar : /usr/man
```

This is a good candidate to be written as:

```
/usr/man willow, cedar, poplar : &
```

Notice that the ampersand substitution uses the whole key string, so if the key in a direct map starts with a "/" (as it should), that slash is carried over, and you could not do something like this:

```
/progs &1, &2, &3 : /export/src/progs
```

This is because Automounter interprets it as follows:

```
/progs /progs1, /progs2, /progs3 : /export/src/progs
```

**Backslash “\”**

Under certain circumstances, you may have to mount directories whose names may confuse Automounter’s map parser. This is a concern only with non-UNIX system servers. An example might be a directory called *rc0:dk1*; this could result in an entry like:

```
/junk -ro vmsserver : rc0 : dk1
```

The presence of the two colons in the location field confuses Automounter’s parser. To avoid this confusion, use a backslash to escape the second colon and remove its special meaning of separator:

```
/junk -ro vmsserver : rc0\ : dk1
```

**Double quotes (“”)**

You can also use double quotes, as in the following example, where they hide the blank space in the name:

```
/smile dentist : /"front teeth"/smile
```

**Environment variables**

Environment variables can be used by prefixing a dollar sign “\$” to the name of the variable. Braces are used to delimit the name of the environment variable from appended letters or digits. These variables can be used anywhere in an entry line, except as a *key*.

Environment variables can either be inherited from the environment or can be defined explicitly with the **-D** command-line option. For instance, if you want each client to mount client-specific files in the network in a replicated format, create a specific map for each client according to its name, so that the relevant line for host *oak* is:

```
/mystuff cypress, ivy, balsa : /export/hostfiles/oak
```

For *willow* it is:

```
/mystuff cypress, ivy, balsa : /export/hostfiles/willow
```

This scheme is viable within a small network, but maintaining this kind of host-specific map across a large network would soon become unfeasible. The solution in this case is to start Automounter with a command line similar to the following:

```
automount -D HOST='hostname' ...
```

The entry in the direct map should read:

```
/mystuff cypress, ivy, balsa : /export/hostfiles/$HOST
```

Now each host finds its own files in the *mystuff* directory, and the task of centrally administering and distributing the maps becomes easier.

## Including other maps

A line of the form `+mapname` causes Automounter to consult the mentioned map as if it were included in the current map. If *mapname* is a relative pathname (no slashes), Automounter assumes it is an NIS map. If the pathname is an absolute pathname, Automounter looks for a local map of that name. If the mapname starts with a dash "-", Automounter consults the appropriate built-in map (a special map that NIS creates by default; see the "Mount point *net*" section in this chapter).

For instance, you can have a few entries in your local *auto.home* map for the most commonly accessed home directories and follow them with the included NIS map:

```
ivy -rw & : /home/&
oak -rw & : /export/home
+auto.home
```

After consulting the included map, Automounter continues scanning the current map if no match is found, so you can add more entries, for instance:

```
ivy -rw & : /home/&
oak -rw & : /export/home
+auto.home
* -rw & : /home/&
```

Finally, as mentioned before, the map included can be a local file or even a built-in map:

```
+auto.home.finance # NIS map
+auto.home.sales # NIS map
+auto.home.engineering # NIS map
+/etc/auto.mystuff # local map
+auto.home # NIS map
+-hosts # built-in hosts map
* & : /export/& # wild card
```

## Starting Automounter

Before starting Automounter, make sure that:

- there are no equivalent entries in */etc/default/filesys*
- all the entries in the maps refer to NFS exported-files

The syntax to invoke Automounter is:

```
automount
 [-mnTv]
 [-D name = value]
 [-f master-file]
 [-M mount-directory]
 [-tl duration]
 [-tm interval]
 [-tw interval]
 [directory map [-mount-options]] ...
```

The **automount(NADM)** manual page contains a complete description of all options. The *mount-options* that you can specify on the command line or in the maps are the same as those for a standard NFS **mount**, excluding **bg** (background) and **fg** (foreground), which do not apply.

By default, the NFS startup script, */etc/nfs*, starts Automounter at boot time through the lines:

```
echo "automount\c"
automount -m -f /etc/auto.master
```

The **-m** option instructs Automounter not to look for the NIS map; the **-f** option instructs it to use the local file, */etc/auto.master*, as the master map.

To use the NIS *auto.master* map, start **automount** without the **-m** option:

```
automount
```

If the map is found, Automounter follows the directives contained within. If NIS is not running or the map is not found, Automounter exits silently.

It is also possible to specify master map information, that is, mount-points, map-names, and mount-options, on the **automount** command line itself:

```
automount /net -hosts /home /etc/auto.home -rw /- /etc/auto.direct -ro
```

This is equivalent to starting **automount** with no options and the following */etc/auto.master* map:

```
/net -hosts
/home /etc/auto.home -rw
/- /etc/auto.direct -ro
```

Other combinations of **automount** arguments can change the name of the master map, and add, nullify, or override master map entries. Again, see the **automount(NADM)** manual page for more details.

## Administering Automounter

---

The following sections describe procedures for Automounter maintenance and debugging.

### Restarting Automounter

---

Every time Automounter mounts or unmounts a file hierarchy, it modifies */etc/mnttab* to reflect the current situation. Automounter keeps an image in memory of */etc/mnttab*, and refreshes this image every time it performs a mounting or an automatic unmounting. If you use the **umount** command to unmount one of the automounted hierarchies (a directory under */tmp/mnt*), you must force Automounter to re-read the */etc/mnttab* file. To do so, enter the command:

```
ps -ef | grep automount | egrep -v grep
```

This gives you the process ID of Automounter. When Automounter receives a SIGHUP signal, it re-reads */etc/mnttab*. To send it that signal, enter:

```
kill -1 pid
```

Here, *pid* stands for the process ID you obtained from the previous **ps** command.

### Modifying the maps

---

You can modify Automounter maps at any time. However, that does not guarantee that all your modifications will take effect the next time Automounter mounts a hierarchy. It depends on what map you modify and what kind of modification you introduce.

**NOTE** Rebooting your machine is generally the simplest way of restarting Automounter.

#### Modifying the master map

Automounter consults the master map only at startup time. A modification to the master map will take effect only next time you reboot the machine.

#### Modifying indirect maps

Entries can be modified, deleted, or added to indirect maps and the change will take effect next time the map is used, which is next time a mount occurs.



## Modifying direct maps

Each entry in a direct map is an automount mount point and it only mounts itself at these mount points at startup. Therefore, adding or deleting an entry in a direct map will only take effect next time you reboot the machine. However, existing entries can be modified (for example, changing mount options or server names, but not names of mount points) while Automounter is running, and will take effect when the entry is next mounted, because Automounter consults the direct maps whenever a mount has to be done.

For instance, suppose you modify the file, */etc/auto.direct*, so that the directory, */usr/src*, is now mounted from a different server. The new entry takes effect immediately (if */usr/src* is not mounted at this time) when you try to access it. If it is mounted now, you can wait until the default unmounting takes place before you access it. If this is not satisfactory, you can unmount with the **umount** command; if you do, you must notify **automount** that the mount table has changed (see the “Restarting Automounter” section earlier in this chapter) before accessing the directory.

You can now mount the directory from the new server. However, if you wanted to delete the entry, you would have to reboot the machine for the deletion to take effect.

**NOTE** For this reason, and because they do not clutter the mount table like direct maps do, indirect maps are preferable and should be used whenever possible.

## Mount point conflicts

---

If you have a home partition on a local disk that is mounted on */home* and you want also to use Automounter to mount other home directories, you may encounter a mount point conflict. That is, if you give it the mount point */home*, then Automounter will hide the local home partition whenever you try to reach it.

The solution is to mount the partition somewhere else, for example, on */export/home*. You would then need an entry in */etc/default/filesys* that says:

```
/dev/xx#z ... /export/home ...
```

Here *xx#z* stands for the name of the partition, assuming that the master file contains a line similar to:

```
/home /etc/auto.home
```

An entry is also needed in the *auto.home* map similar to the following (where *terra* is the name of the machine):

```
terra terra:/export/home
```

If the partition is set up such that home directories are located on */home/machine/user*, move all the directories at the *user* level one level up, so as to eliminate the *machine* level:

```
cd /home
mv machine/* .
rmdir machine
```

You need not change the */etc/passwd* entry for users. Their home directories are still accessible through */home/machine/user*, as before. Instead of doing a mount, Automounter recognizes that the filesystem is on the same machine and establishes a symbolic link from */home/machine* to */export/home*.

## Unexpected filesystem behavior

---

Automounter's use of symbolic links may produce unexpected effects. For example, Automounter's method of mounting a filesystem onto the pseudo-staging area */tmp\_mnt/...* causes undesired results when changing directories using *".."* in a pathname. If the user issues the command *cd ..* from the mount point root, they end up in */tmp\_mnt*, not in the desired parent directory. Similarly, *pwd* also shows the staging area.

This does not affect the Korn shell, which keeps its own copy of the absolute pathname.

## Increasing mount table entries

---

In a system with a high degree of Automounter activity, you must ensure that your filesystem kernel parameters are configured adequately for the volume of mount requests. If you are managing more than eight mounts (including default, NFS and Automounter mounts), you should probably increase the number of entries allocated in the mount table, */etc/mnttab*. When the maximum number of entries is exceeded, Automounter fails and the desired filesystems are not mounted as expected.

You can increase the number of mount table entries by tuning the kernel parameter *NMOUNT*. To do so, follow these steps:

1. After informing users to log off, reboot your system and enter system maintenance mode.
2. Change directories (*cd*) to */etc/conf/cf.d*.

3. Enter the following command to determine the current value of the NMOUNT kernel parameter (the default is 8):

```
./configure -y NMOUNT
```

4. Change the value of NMOUNT with the following command:

```
./configure NMOUNT=new_value
```

The value should be large enough to accommodate your largest volume of mount requests; however, you should keep the value as low as possible to avoid unnecessary search time. You will probably have to experiment to find an optimal value.

5. You must link the new parameter into the kernel. To do so, enter the following command:

```
./link_unix
```

Once the kernel has been relinked, you must reboot your system.

For more information, consult the **configure(ADM)** manual page and the “Tuning system parameters” chapter of the *Administering Operating System Services* guide.

## Debugging Automounter

---

The following sections describe general error messages and messages generated when running the verbose option (**automount -v**). You should run the verbose option if Automounter fails to start properly.

All Automounter error messages are prefixed by the string “automount: ”.

### General error messages

pathok: couldn't find devid *device id*

This is an internal Automounter error.

WARNING: default option '*option*' ignored for map *mapname*

Here *option* is an unrecognized default mount option for the map *mapname*.

WARNING: *option* ignored for *key* in *mapname*

Automounter detected an unknown mount option. This is advisory only. Correct the entry in the appropriate map.

bad entry in map *mapname*

map *mapname*, key *key*: bad

The map entry is malformed, and Automounter cannot interpret it. Recheck the entry; perhaps there are characters in it that need escaping.

can't get my address

Automounter cannot find an entry for its host in */etc/hosts*(or *hosts.byname*).

cannot create UDP service

Automounter cannot establish a UDP connection.

svc\_register failed

Automounter cannot register itself as an NFS server. Check the kernel configuration file.

couldn't create *pathname: reason*

Here *pathname* is */tmp\_mnt* or the argument to the -M command line option.

can't mount *mountpoint: reason*

Automounter could not mount its daemon at *mountpoint*.

exiting This is an advisory message only. Automounter has received a SIGTERM (has been killed) and is exiting.

*server:pathname* no longer mounted

Automounter is acknowledging that *server:pathname*, which it mounted earlier, has been unmounted by the *umount* command. Automounter notices this within 1 minute of the unmount or immediately if it receives a SIGHUP.

trymany: servers not responding: *reason*

No server in a replicated list is responding. This may indicate a network problem.

host *server* not responding

Automounter attempted to contact *server* but received no response.

mount of *server:pathname* on *mountpoint: reason*

Automounter failed to do a mount. This may indicate a server or network problem.

hierarchical mountpoints: *pathname1* and *pathname2*

Automounter does not allow its mount points to have a hierarchical relationship; that is, an Automounter mount point must not be contained within another automounted filesystem.

*mountpoint*: Not a directory

Automounter cannot mount itself on *mountpoint* because it is not a directory. Check the spelling and pathname of the mount point.

dir *mountpoint* must start with "/"

The Automounter mount point must be given as the full pathname. Check the spelling and pathname of the mount point.

*mapname*: yp\_err

There is an error in looking up an entry in an NIS map. It may indicate NIS problems.

*hostname* : exports: rpc\_err

There is an error getting export list from *hostname*. This indicates a server or network problem.

nfscast: Cannot send packet: *reason*

Automounter cannot send a query packet to a server in a list of replicated filesystem locations.

nfscast: cannot receive reply: *reason*

Automounter cannot receive replies from any of the servers in a list of replicated filesystem locations.

nfscast:select: *reason*

Cannot create socket for nfs: rpc\_err

These error messages indicate problems attempting to reach servers for a replicated filesystem using the ping command. This may indicate a network problem.

## Error messages generated by the verbose option

no mount maps specified

Automounter was invoked with no maps to serve, and it cannot find the NIS *auto.master* map. It exits. Recheck the command, or restart NIS if that was the intention.

*mapname*: not found

The required map cannot be located. This message is produced only when the -v option is given. Check the spelling and pathname of the map name.

leading space in map entry *entry text* in *mapname*

Automounter discovered an entry in a map that contains leading spaces. This is usually an indication of an improperly continued map entry. For example:

```
foo /bar frobz:/usr/frotz
```

In this example, the warning is generated when Automounter encounters the second line, because the first line should be terminated with a backslash "\".

bad key *key* in indirect map *mapname*

While scanning an indirect map, Automounter found an entry key containing a "/". Indirect map keys must be simple names, not pathnames.

bad key *key* in direct map *mapname*

While scanning a direct map, Automounter found an entry key without a prepended "/". Keys in direct maps must be full pathnames.

NIS bind failed

Automounter cannot communicate with the ypbind daemon. This is information only. Automounter continues to function correctly provided it requires no explicit NIS support. If you need NIS, check to see if there is a ypbind daemon running.

couldn't create mntpnt *mountpoint*: *reason*

Automounter cannot create a mount point required for a mount. This usually occurs when attempting to mount hierarchically all of a server's exported filesystems. A required mount point may exist only in a filesystem that cannot be mounted (it may not be exported) and it cannot be created because the exported parent filesystem is exported read-only.

WARNING: *mountpoint* already mounted on

Automounter is attempting to mount over an existing mount point. This indicates an Automounter internal error.

*server.pathname* already mounted on *mountpoint*

Automounter is attempting to mount over a previous mount of the same filesystem. This could happen if an entry appears both in */etc/default/filesys* and in an Automounter map (either by accident or because the output of **mount -p** was redirected to *default/filesys*). Delete one of the redundant entries.

can't mount *server.pathname*: *reason*

The mount daemon on the server refuses to provide a file handle for *server.pathname*. Check the export table on *server*.

remount *server.pathname* on *mountpoint* : server not responding

Automounter failed to remount a filesystem it previously unmounted. This message may appear at intervals until the filesystem is successfully remounted

WARNING: *mountpoint* not empty!

The mount point is not an empty directory. The directory *mountpoint* contains entries that are hidden while Automounter is mounted there. This is advisory only.

## Chapter 12

# LAN Manager client administration

---

This chapter describes procedures that are used in administering a LAN Manager client. Client administration includes the following tasks:

- adding a client to the network
- configuring a client
- starting and stopping the client
- obtaining client status

The LAN Manager client can connect to resources on the following servers:

- LAN Manager for UNIX Systems Server
- XENIX-NET Server
- Microsoft LAN Manager Server for OS/2
- Microsoft MS-NET Server

## Understanding NetBIOS

---

The NetBIOS interface allows LAN Manager to send messages over different transports. Each time LAN Manager uses NetBIOS for a particular purpose, it first establishes a NetBIOS session. Two LAN Manager components use NetBIOS sessions:

- connections to resources on a server
- utilities (such as `vtp`)

Internal tables keep track of these sessions. You can configure the number of NetBIOS sessions available, but the maximum and default number of sessions vary with the NetBIOS software used.



## Configuring NetBIOS to run with TCP/IP

---

NetBIOS has special considerations when running with TCP/IP as described in the following sections. For example, the format of the */etc/hosts* file can vary from one vendor's TCP/IP to another. If you modify this file, be sure to follow the correct format of that machine. The way broadcast messages are handled can also vary from one vendor's TCP/IP to another. Investigate any differences among the various versions of TCP/IP on your network and the methods for resolving those differences. Finally, be sure that the NetBIOS scope identifier has the same value on all the computers running NetBIOS and TCP/IP on the same network.

### Host files

Some implementations of TCP/IP have slightly different formats for the */etc/hosts* file, such as the *localhost* keyword. In the SCO implementation of TCP/IP, the *localhost* keyword is specified on the same line as the loopback IP address:

```
127.0.0.1 localhost loopback lb
```

Other implementations of TCP/IP specify the *localhost* on the same line as the name of the local host's IP address. Because of these differences, only copy a master */etc/hosts* file to all computers on the network if they all run the same implementation of TCP/IP. Refer to the documentation accompanying your TCP/IP transport for more information.

### Broadcasts

Broadcasts allow TCP/IP to send certain types of messages, such as alerts, to a wide audience on a LAN. NetBIOS uses broadcasts for tasks such as name registration.

The following steps will help ensure that broadcasting is working correctly, with the assumption that TCP/IP and NetBIOS are running on both the local and remote machines. It should be possible to run the local form of the *nbstatus* command on the local client machine, and also on the remote machine if the *nbstatus* command is available on that machine. The *nbstatus* command with no arguments should return NetBIOS status information if NetBIOS is running on the local machine.

1. Verify that the network and subnet portions of the IP address are the same on all computers. Broadcasts do not repeat across networks or subnets, so the network and subnet addresses must be identical. The configuration of IP addresses varies according to the specific implementation of TCP/IP. In SCO TCP/IP, the local IP address is configured on the *ifconfig* command line in the */etc/tcp* file. You must reboot the machine so that the changes

made to */etc/tcp* file take effect. To configure the IP address in another implementation of TCP/IP, refer to the documentation for that implementation.

2. Verify that all computers on the network agree on either all 0's or all 1's for the broadcast address. Using all 1's is recommended. In SCO TCP/IP, the broadcast address is specified on the **ifconfig** command line in the */etc/tcp* file by using 255 for each byte of the host portion of the IP address. For example:

```
ifconfig e3A0 132.147.64.1 broadcast 132.147.64.255 netmask 255.255.255.0
```

The network portion of the address in the preceding example is 132.147.64, and the host portion is set to all 1's by appending 255 to the network portion. If you change the */etc/tcp* file, you must reboot before the changes take effect.

3. Verify that all computers agree on the broadcast address relative to the subnet mask. Some implementations of TCP/IP ignore the subnet portion of class C IP addresses. If this is a problem, replace the subnet number with the broadcast number when defining the broadcast address. Using the same example as in step 2, where the subnet portion is 64, the broadcast address would be configured as follows:

```
ifconfig e3A0 132.147.64.1 broadcast 132.147.255.255 netmask 255.255.255.0
```

If you change the broadcast address, reboot the machine to have the changes take effect.

4. Verify that NetBIOS is working by using the form of the **nbstatus** command that requests status from a remote computer. For example, to verify that NetBIOS is working on the machine on your network named *fred*, enter the command:

```
nbstatus FRED
```

Note that the computername is specified in all uppercase letters. If NetBIOS is functioning between the two machines, a list of information about the remote computer will be displayed. If an error is returned, this indicates that NetBIOS is not working, either because there is still a TCP/IP broadcast incompatibility, or because different NetBIOS scope identifiers are being used (see "Setting the scope identifier"). Note that it is sometimes possible for the remote form of the **nbstatus** command to return an error under normal circumstances. Some implementations of NetBIOS do not accept the remote **adapter status** NCB; also, if the remote machine is a LAN Manager for UNIX Systems client, and there currently are no sessions between the client and a server, then NetBIOS status is not returned.

### *Setting the scope identifier*

The NetBIOS scope identifier is a string of characters used when network applications communicate using NetBIOS over TCP/IP. The NetBIOS scope identifier must be the same for all computers on a NetBIOS network using TCP/IP as the transport. If you have different scope identifiers on your computers, NetBIOS assumes they are part of a different logical network, and cannot connect to them.

Before setting the NetBIOS scope identifier, check the value of the NetBIOS scope identifier on the other machines in the network. If you have MS-DOS or OS/2 workstations on the network, the NetBIOS scope identifier should be set to NULL on all machines on the network. Many implementations of NetBIOS for TCP/IP on MS-DOS and OS/2 LAN Manager allow you to use only the NULL scope identifier. If you have any other LAN Manager-compatible machines on the network, check their documentation for restrictions.

The scope identifier's setting depends on the implementation of TCP/IP. SCO TCP/IP sets the scope identifier with the `NB_SCOPE` parameter in the file `/etc/default/nbconf`. For example, to set the scope identifier to NULL, follow these steps:

1. Log in as *root*.
2. Go to the shell prompt.
3. If LAN Manager client is running, enter the **net stop client** command.
4. Stop NetBIOS using the **netbios stop** command.
5. Use a text editor to change the line containing the `NB_SCOPE` parameter in the `/etc/default/nbconf` file to `NB_SCOPE=""`.
6. To activate the change to the `/etc/default/nbconf` file, either reboot your machine or enter the command **netbios start**.
7. Type **net start client**, unless you rebooted in step 6 and have your client configured to start automatically.

For other implementations of TCP/IP, check the accompanying documentation for information on how to set the scope identifier.

To verify that NetBIOS is working with the new scope, follow step 4 in the preceding section, "Broadcasts."

### *NetBIOS keepalives*

NetBIOS keepalives can be useful for determining whether or not machines that you are connected to are still active, at the cost of a small amount of extra network activity. For instructions on turning on keepalives, see the "Kernel configuration" section of the "TCP/IP network administration" chapter in this guide.

## *user ID consistency*

---

Many functions of the network depend upon a consistent user identification (user ID) number for each user on all machines on the network. These user names and user ID are located in the `/etc/passwd` file. If you are connecting existing computers in a network for the first time, be certain that user IDs are consistent from machine to machine. You may need to retire some user IDs on existing machines and create new ones for network compatibility. Use `sysadmsh(ADM)` to define or retire a user ID. Refer to the manual page entries for more information about `passwd(C)` and the `/etc/passwd` file.

## *Planning for the virtual terminal program*

---

Before you set up users for accessing the virtual terminal program, `vtp(LM)`, these users must have accounts on the LAN Manager for UNIX System or XENIX-NET Servers they need to access. Be sure each user has the same user ID and group ID on all accounts on the network. To support virtual terminals, the server must be running `svrdaemon(LM)`.

Make sure enough NetBIOS sessions are configured on both the client and the server. For more information, refer to the section "Managing NetBIOS sessions" later in this chapter.

## *Accessing a server with user-level security*

---

In a LAN Manager network that has LAN Manager for UNIX Systems Clients and servers with user-level security, the network administrator has two options for limiting access to server resources from those clients: require all users on all LAN Manager clients to know a single network-wide password or require each user on a LAN Manager client to know their own, probably different, password. In the first case, the network administrator ensures that the LAN Manager guest account is set up on the server and tells every LAN Manager client user what the password is for the guest account. In the second case, the network administrator sets up a user account on the server for every machine on the network running a LAN Manager client. These accounts (client machine accounts) must bear the network names of the machines, not the names of the users working at those machines. The network administrator can then assign a unique password to each of these accounts and tell each individual user what the password is for their machine.

## *Adding and removing a client*

---

To add or remove a client, run the **custom(ADM)** program according to the installation instructions in the SCO system *Installation and Upgrade Guide*.

To change the name of the client, use a text editor and edit the */etc/conf/cd.f/stune* file. Make sure this name is consistent with the name in the */etc/systemid* file. To check the name of the client, enter:

```
net name
```

at the UNIX System prompt.

To check the contents of */etc/systemid*, enter the following at your prompt:

```
uname -s
```

## *Configuring the client for user access to servers*

---

Configuration of the LAN Manager client consists mostly of cross-checking information with servers. Make sure that the users on LAN Manager clients have the necessary names and passwords for the resources that they need:

- Users accessing shared resources on a LAN Manager server with share-level security need to know the names and passwords of those resources.
  - Users accessing user accounts on a LAN Manager server with share-level security (via autoshare) need to know the names and login passwords of those accounts.
  - Users accessing a LAN Manager server with user-level security need to know the password for either the LAN Manager guest account or their client machine account, depending on which account is to be used. They also need to know the names of any shared resources or user accounts to which they are to have access.
  - Users accessing XENIX-NET servers with trusted access need to have the name of the server in their */usr/lib/lm/conspw* file, and to be configured on the XENIX-NET server using **mkcon**.
- Users of LAN Manager client that access XENIX-NET Servers using TRUSTED ACCESS have the same permissions on the XENIX-NET Server as they do on the LAN Manager client. They can establish sessions on the server without the **net use** command. The SCO system *User's Guide* provides instructions on how to access a XENIX-NET server from a LAN Manager client using trusted access. The XENIX-NET documentation describes how to set up a XENIX-NET Server for trusted access.

All XENIX-NET Servers to which a LAN Manager client has trusted access must be listed in the */usr/lib/lm/conspw* file of the client. Make sure that each user has the same user ID and group ID on the LAN Manager client and the XENIX-

NET server. After modifying the `/usr/lib/lm/conspw` file on the LAN Manager client, you must stop and restart the client for the modifications to take effect. In addition, you must define the LAN Manager client on the XENIX-NET Server using the `mkcon` command.

The following example shows a `conspw` file for a LAN Manager client with trusted access to the XENIX-NET servers *cherry*, *host2*, and *host3*:

```
cherry:mypasswd
host2:secret
host3:hidden
```

It is easier to administer the network if each user has the same accounts, passwords, user IDs, and group IDs on all machines.

You must reboot the system to have changes made to the `conspw` file take effect.

## Client operating parameters

The operating parameters for the client are defined in the `/usr/lib/lm/lmxrc` file. During installation, default values are assigned to these parameters. For information on how to modify these defaults, refer to the section “Parameters affecting connections to servers: `lmxrc`” later in this chapter.

## Starting and stopping the client

To start the client, enter `net start client` at the shell prompt. When the client starts, it reads the `lmxrc` file to set up its operating environment. For more information on the `lmxrc` file and how to modify it, refer to the section “Modifying client parameters” later in this chapter.

To stop the client, enter `net stop client`. The `net stop all` command stops any LAN Manager software that has been started. To stop all sessions between your client and servers to which you are connected, enter the command `net abort`.

Optionally, you can configure LAN Manager client so that the client starts automatically when the system is booted. If you did not configure LAN Manager client to start automatically when you installed it, you can do so now by copying two files from the `/usr/lib/lm/rc` directory as follows:

```
cp /usr/lib/lm/rc2.d/S91lmcx /etc/rc2.d
cp /usr/lib/lm/rc0.d/K66lmcx /etc/rc0.d
```

To disable automatic startup of LAN Manager client, remove the files `/etc/rc2.d/S91lmcx` and `/etc/rc0.d/K66lmcx`.

## Managing the client

---

The following commands help you manage the client:

- The **net stat client** command returns information about client sessions.
- The **nbstatus** command provides NetBIOS status and information.

### Client status

---

The **net stat client** command returns information that resembles the following screen:

```
1.1.1 LAN Manager Software installed on node kiwi.

Client enabled. Fri Aug 3 16:19:43 1990

Consuming from VC State dialect vcid nprocs ntrans bufisz
avocado dormant * Core 1 0 82 3k
 b (owner root) connected to server avocado
```

The banner shows the version of your LAN Manager client and the current time and date. The rest of the status is about network activity at the SMB protocol level. The different columns contain the following information:

**Consuming from**

name of the server that is being accessed from your client

**VC State**

tells whether a virtual circuit between the client and the server is active. The following are possible VC States:

- **aborted** means that an error has occurred on this virtual circuit. Usually this circuit is automatically re-used. Sometimes, however, it is necessary to use the **net abort(LM)** command to reset the circuit.
- **active** means that the virtual circuit is currently in use; that is, the client has a file or directory open on the server.
- **dormant** means that the virtual circuit is open and ready for use, but is not currently in use by the client.

**dialect**

which dialect of the SMB protocol is being used over the virtual circuit

**vcid**

ID number of a particular virtual circuit.

**nprocs**

number of client processes with open files or directories on the server

**ntrans**

number of request/reply transactions made across the virtual circuit

## Checking NetBIOS status

The **nbstatus** command provides NetBIOS status from LAN Manager client or from a remote LAN Manager server. LAN Manager client must be started and have at least one connection to a server before NetBIOS status can be obtained.

The output from **nbstatus** resembles the following display:

```
Adapter status for adaptor 0, name *:
ID: 08 00 14 51 62 09 (655364) Jumpers 0 Selftest 0 Version 1.0
Time up 5:51
Packets sent: 2176 CRC errors: 0
Packets received: 99060 Alignment errors: 0
Retransmitted packets: 0 Collisions: 0
Resources exhausted: 0 Aborted transmissions: 0
Ncbs: Free : 45 of 48; maximum configurable: 48
Sessions: In use: 2 of 48; maximum configurable: 48
Adapter packet size: 2048
Local name table (2 names):
Name Soc Num Status
KIWI x00 3 0x4

Session status for adaptor 0, name *:
name num 255, # sessions 3, rcv dg=1, rcv any=0
LSN State Local Name Soc Remote Name Soc rcvs sends
1 ESTAB KIWI x00 CHERRY x20 0 0
2 ESTAB kiwi x73 cherry x73 1 0
```

The information in the first part of the display shows NetBIOS activity and gives information useful for troubleshooting at the NetBIOS level. If there is noise on the line, for example, NetBIOS may have several aborted transmissions or retransmitted packets. Additional information appears if the client is active.

The next item, the local name table, shows the names that have been added to NetBIOS. The fields are as follows:

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | name in use                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Soc  | numeric value of the 16th character of the name. By convention, uppercase names are used in the SMB filesharing protocol, and lowercase names are used by other LAN Manager utilities. The significance of the Soc field is defined by the NetBIOS application. The SMB protocol uses this field to indicate a particular service or function. For example, the LAN Manager client uses the machine name in uppercase with a Soc value of 0x00 (ASCII code for the null character) to connect to the remote filesharing server. |



|        |                                                                                                                         |
|--------|-------------------------------------------------------------------------------------------------------------------------|
| Num    | number assigned to the name by NetBIOS when the name was added to the table                                             |
| Status | status of the name that NetBIOS has added or is adding to the name table. Valid numbers have the following definitions: |
| 0x0    | NetBIOS is trying to register a name in the local name table                                                            |
| 0x4    | NetBIOS has successfully registered this name in the local name table                                                   |
| 0x5    | NetBIOS has de-registered this name from the local name table                                                           |
| 0x6    | NetBIOS has detected a duplicate name in the local name table                                                           |
| 0x7    | NetBIOS has detected a duplicate name with de-register pending                                                          |
| 0x84   | name is a group name                                                                                                    |

The last item in the nbstatus output is the session status table. The header of this section shows the number of session table entries used, including NetBIOS LISTEN's, and the number of RECEIVE DATAGRAMS and RECEIVE ANY NCB's that are posted to the NetBIOS. The session status table shows the following information:

|             |                                                        |
|-------------|--------------------------------------------------------|
| LSN         | NetBIOS local session number (LSN) for the session     |
| State       | state of the session. Possible states are:             |
| LISTEN      | Waiting for a connection                               |
| CALL        | Attempting to connect                                  |
| ESTAB       | An established connection                              |
| CLOSING     | Connection is in the process of closing                |
| CLOSED      | Connection is closed                                   |
| ABORT       | Connection has been aborted                            |
| Local Name  | local name associated with the session                 |
| Soc         | numeric value of the 16th character of the name        |
| Remote Name | server name associated with the session                |
| Soc         | numeric value of the 16th character of the server name |
| rcvs        | number of RECEIVE NCB's outstanding on the session     |
| sends       | number of SEND NCB's outstanding on the session        |

The **nbstatus** display indicates how many NCBs (Network Control Blocks) NetBIOS is using. The following line, for example, shows that three NCBs are in use:

```
Ncbs: Free : 45 of 48; maximum configurable: 48
```

To verify this number, look at the session status table. There are one RECEIVE DATAGRAM (rcv dg=1) NCB in use, one RECEIVE NCB pending in the rcvs column, and one NCB taken by the **nbstatus** command itself.

NetBIOS establishes one session to a server for connecting to resources, and additional sessions for **vtp** and mail going over the NetBIOS channel.

To obtain adapter status from remote machines, you can use the **nbstatus MACHINE** command. **MACHINE** is the name of the remote machine in all uppercase letters. On a client using coresident transports, the remote adapter status query is issued on the primary NetBIOS driver (adapter 0) by default. To obtain information from a machine located on the second NetBIOS network, you must specify adapter 1 in the **nbstatus** command line. For example, the command

```
nbstatus -n 1 FRED
```

will report the status on machine *fred* on the second NetBIOS network. For more information on using **nbstatus**, see the section “Checking NetBIOS status” in this chapter and the **nbstatus** manual page.

## Utility administration on the client

---

The **vtp** utility, a terminal emulation program that allows users to log in to remote UNIX systems, is part of LAN Manager client. This utility is automatically installed when you install LAN Manager client according to the instructions in the SCO sytem *Installation and Upgrade Guide*.

For this utility to function properly, it must be configured both on your LAN Manager client and on the appropriate server.

### Configuring the client for vtp

---

**vtp** allows a user on a LAN Manager client to log in to a LAN Manager for UNIX Systems or XENIX-NET Server as a remote user. To use **vtp** on the LAN Manager client, the following requirements must be met on the server:

- The server must be running **srvdaemon** and the **vtprsv** service must be defined to the daemon. For more information, see the administrator’s documentation for your server.
- The user must have an account on the server.

## Modifying client parameters

---

The LAN Manager client uses several files to track operating parameters. These parameters are set to defaults during installation, but you can alter them to create the best operating environment for your network. When you alter parameters, be aware that the parameters in one file might affect the parameters in another file. When this is the case, it is noted in the description of that parameter.

Remember, changes to most of these parameters do not take effect immediately. You must either stop and restart LAN Manager client, or reboot the system.

The following files contain operating parameters for LAN Manager client:

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>constable</i>    | contains parameters that affect data transfer. The parameters in this file take effect when the client is started. The <i>constable</i> file is in the <i>/usr/lib/lm</i> directory.                                                                                                                                                                                                                                                                                                                                                 |
| <i>lmxrc</i>        | contains parameters that take effect when the client is started. The <i>lmxrc</i> file is in the <i>/usr/lib/lm</i> directory.                                                                                                                                                                                                                                                                                                                                                                                                       |
| <i>mtune, stune</i> | contain parameters that are used by the UNIX system kernel to allocate data segments. Some of these kernel parameters apply to the operating environment of LAN Manager client. Changes to these parameters take effect when the kernel is relinked, and the UNIX System is rebooted. These parameters affect the client. The <i>mtune</i> and <i>stune</i> files are in the <i>/etc/conf/cf.d</i> directory. See the <i>Administering Operating System Services</i> section for more information on <i>mtune</i> and <i>stune</i> . |

## Managing NetBIOS sessions

---

LAN Manager client uses internal tables to track NetBIOS sessions. The sizes of these tables are set during the installation, but they can be modified by adjusting the parameters in one or more files. Two components of LAN Manager use NetBIOS sessions:

- LAN Manager client connections to servers
- utilities, such as *vtp*

The following parameters control the number of NetBIOS sessions:

|                    |                                                                                                                                                                                                                        |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MAXSESSIONS</b> | This parameter, configured in the <i>mtune</i> file, limits the total number of NetBIOS sessions allowed by all components of LAN Manager. Any component can use all available NetBIOS sessions, but you cannot assign |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

session priority to any one component. Remember to keep the number of MAXSESSIONS equal to or less than the maximum number allowed by the NetBIOS driver.

For more information on setting TCP/IP NetBIOS driver parameters, see the section “Kernel configuration” in the “TCP/IP network administration” chapter of this guide.

|                       |                                                                                                                                                                                                                                                                   |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MAX_CONFS_SESS</b> | This startup parameter, configured in the <i>lmxrc</i> file, sets the maximum number of NetBIOS sessions that LAN Manager client can use. This parameter does not affect the number of sessions that can be used by either a server or the LAN Manager utilities. |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

If you set **MAX\_CONFS\_SESS** higher than **MAXSESSIONS**, you see an error message when the client is started. To ensure that sessions remain for other LAN Manager components, set **MAX\_CONFS\_SESS** lower than **MAXSESSIONS** to limit the number of NetBIOS sessions that LAN Manager client uses.

## Modifying the client for performance: constable

This section explains how to improve LAN Manager client performance by altering the configuration of your system. The following modifications are not required for normal operation of LAN Manager client.

### Adjusting client read and write windows

By editing the file `/usr/lib/lm/constable`, you can adjust the read and write windows for each server to which LAN Manager connects. You can enhance network performance by carefully tuning these parameters. Stop and restart the client to activate these changes.

The *constable* file has the following format:

|        | Special<br>Read Support | General<br>Read-Ahead |        |       | General<br>Write-Behind |
|--------|-------------------------|-----------------------|--------|-------|-------------------------|
| server | reado                   | reada                 | read   | read  | write                   |
| name   | on/off                  | timeout               | window | ahead | window                  |
| *      | 1                       | 500                   | 3      | 3     | 4                       |

Note that every line in the file except the last one is commented out. This line has an asterisk “\*” in the “server name” column, which means all servers, and

the line lists the default values for each parameter. To specify different values for one server, add a line to the bottom of the file containing the server's name at the beginning of the line, and specify the values you want to use for that server in the other columns. Any machine without a line of its own uses the defaults specified on the line beginning with the asterisk.

The following list describes the parameters and modifications for improved performance.

- |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>reado on/off</b>  | This parameter is a toggle. A value of 1 turns on an improved protocol set, and a value of 0 turns it off. It should be set to 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>reada timeout</b> | <p>This parameter sets the length of time that buffered data remains valid, in units of 1/50th of a second. For example, if this parameter is set to 500, data read from a server and placed in a buffer remains valid for 10 seconds. If the data is not read from the buffer within this period, the buffer is flushed and the data is re-read from the server.</p> <p>Increasing the value of this parameter improves network performance, but it can cause data integrity problems if the data read change frequently. As a guide, use a high value for servers that contain relatively stable data (such as archive servers) and a lower value for servers with more volatile data.</p> |
| <b>read window</b>   | This parameter specifies how many requests can remain outstanding before a reply is received. A value of zero or one means that after a request is sent, no more requests are sent until a reply is received. A value of 2 allows two requests to be sent before a reply is received. Figure 12-1 illustrates this difference.                                                                                                                                                                                                                                                                                                                                                               |

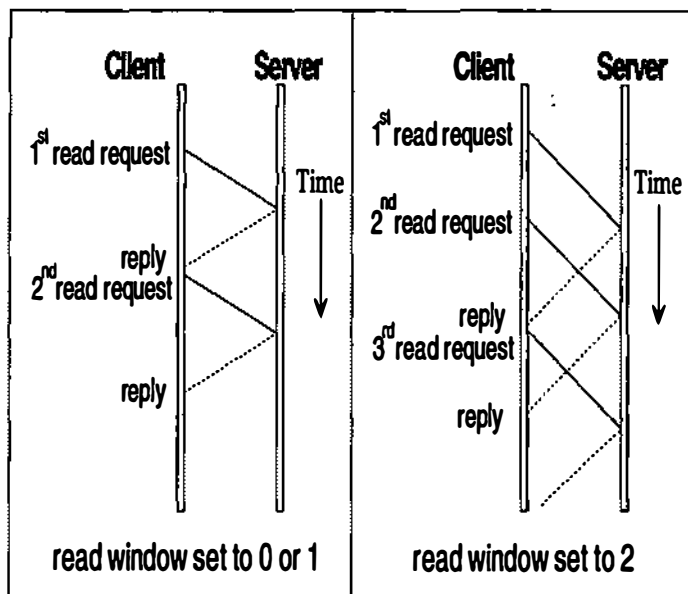


**Figure 12-1 The read window size affects performance**

Although setting this parameter to a high value improves network performance, it drains system resources. You may wish to test several different values to find what works best on your system.

#### **read ahead**

The read-ahead value is the number of extra read requests that LAN Manager client sends after the initial request is satisfied. By setting this parameter to a value greater than 0, LAN Manager client can anticipate a user process by requesting data for which the process has not yet asked.

A high read-ahead value improves network performance, but it drains system resources. As with the read window, the best way to find the ideal value is through experimentation.

#### **write window**

The write window resembles the read window. This parameter specifies the number of write requests that can be outstanding before the server acknowledges that the data was received intact. This parameter is effective only when accessing a XENIX-NET 1.3 Server.

## *Adjusting network buffer sizes*

The network buffer size is the maximum amount of data that can be sent from or received by a machine in a single packet. You can improve network performance by properly configuring buffer sizes.

The following guidelines should be used for adjusting buffer sizes:

- Large buffers use more memory. If you specify too large a buffer size, other processes may lack memory.
- When two machines with different buffer sizes are involved in a transaction, the smaller buffer size is used. This is called the “effective buffer size.”

To find the network buffer size for LAN Manager client communications with a server, enter the **net stat client** command on the client. The “bufsz” column of the **net stat client** output lists the effective buffer size for every machine to which the client is connected.

The following two parameters in the file */usr/lib/lm/lmxcrc* specify the size and number of the network buffers:

- **NETBUFSIZE** configures the buffer size.
- **NETBUFFERS** configures the number of buffers.

If these parameters are set to zero, LAN Manager client checks the amount of available memory and sets **NETBUFSIZE** and **NETBUFFERS** to reasonable values. To override the autoconfigured default, set the parameter to a non-zero value.

The maximum allowed size for the network buffers is determined by the size of the largest Streams buffer configured on the SCO system. Use the following formula to calculate the maximum allowable size:

$$\text{Max Network Buffer size} = (\text{Max Streams Buffer Size} * 2) - 48 \text{ bytes}$$

The preceding formula yields the theoretical maximum buffer size, although the size of the network buffers allocated must be a whole number multiple of 1 Kbyte.

To determine the maximum Streams buffer size on your machine, check the pertinent parameters in the file */etc/conf/cf.d/mtune*. Table 12-1 lists these parameters, together with minimum and maximum allowed values. The first column also lists the default values for these parameters if no override values exist in the companion file, */etc/conf/cf.d/stune*. All changes to the default values must lie between the minimum and maximum values and be recorded in the *stune* file. See the section “Modifying kernel parameters: *mtune*” later in this chapter for additional information.

**Table 12-1 Stream buffers defaults and limits**

| Parameter | Default | Minimum | Maximum |
|-----------|---------|---------|---------|
| NBLK4096  | 0       | 0       | 2048    |
| NBLK2048  | 20      | 0       | 2048    |
| NBLK1024  | 20      | 0       | 2048    |
| NBLK512   | 8       | 0       | 2048    |
| NBLK256   | 8       | 0       | 2048    |
| NBLK128   | 8       | 0       | 2048    |
| NBLK64    | 40      | 0       | 2048    |
| NBLK16    | 40      | 0       | 2048    |
| NBLK4     | 40      | 0       | 2048    |

The parameters in Table 12-1 set the number of Streams buffers of various sizes. A machine with these default values has no 4-Kbyte buffers configured, twenty 2-Kbyte buffers, twenty 1-Kbyte buffers, eight one-half Kbyte buffers, and so on. Because no 4-Kbyte buffers are configured, the Maximum Streams buffer size on this machine is two Kbytes, or 2048 bytes. Thus, the maximum computed network buffer size is 4048 bytes. Rounding down to the nearest whole number of Kbytes yields a maximum allocated network buffer size of three Kbytes. If the default value of the NBLK4096 parameter in *mtune* is overridden in the *stune* file to a non-zero value, thus configuring one or more 4-Kbyte Streams buffers, the maximum network buffer size is 7 Kbytes.

## *Parameters affecting connections to servers: lmxrc*

The parameters located in the */usr/lib/lm/lmxrc* file determine various functional limits in LAN Manager client software. LAN Manager client reads the *lmxrc* file when it is started and checks for entries that override default values for these parameters.

The network administrator can change some network parameters by modifying the */usr/lib/lm/lmxrc* file. Parameters usually do not need to be changed, and such modifications can adversely affect performance.

The following table shows the parameters and default values in the */usr/lib/lm/lmxrc* file. Parameter descriptions follow the table.

**Table 12-2 lmxrc parameters and defaults**

| Parameter      | Default Value |
|----------------|---------------|
| NETBUFSIZE     | 0             |
| NETBUFFERS     | 0             |
| MAX_CONFS_SESS | 15            |
| UC_SMBPATH     | 0             |



|                       |                                                                                                                                                                                                                                                         |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NETBUFSIZE</b>     | defines the size of the buffer space used to send and receive messages in bytes. If a value of 0 is specified, the value of this parameter is determined automatically. The default is 0.                                                               |
| <b>NETBUFFERS</b>     | specifies the number of network buffers allocated. It is configured automatically if a value of 0 is specified. The default is 0.                                                                                                                       |
| <b>MAX_CONFS_SESS</b> | specifies the number of virtual circuits for use by the client. Refer to the <b>MAXSESSIONS</b> parameter in the section "Managing NetBIOS sessions" and the next section in this chapter. The default value is 15.                                     |
| <b>UC_SMBPATH</b>     | indicates whether all pathnames in an SMB are to be specified in uppercase, converting lowercase letters to uppercase as needed. A value of 0 indicates no conversion is to take place. A value of 1 indicates conversion will occur. The default is 0. |

### *Modifying kernel parameters: mtune*

---

Some of the parameters that LAN Manager uses are defined in kernel space. These parameters are in the *mtune* and *stune* files in the */etc/conf/ckf.d* directory. When you modify a kernel parameter, the UNIX system makes a copy of the *mtune* parameter and puts the modified value of the parameter in *stune*. Although the default version of all kernel parameters is always in *mtune*, the value of any kernel parameter in *stune* overrides the value in *mtune*.

The *stune* file is automatically created or modified when you install and configure LAN Manager and when you use **configure(ADM)** and **idtune(ADM)** to modify kernel parameters. Use **configure** to modify parameters that come with the system. Use **idtune** to modify parameters that are added when LAN Manager client is installed. You must relink the kernel and reboot the system before changes to kernel parameters can take effect.

When LAN Manager client is configured, kernel parameters are increased to accommodate the client. The following parameters already exist in *mtune* before LAN Manager client is installed but are modified in *stune* when LAN Manager client is configured:

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NBLK<sub>n</sub></b> | <b>NBLK16</b> through <b>NBLK4096</b> control the number of Streams data blocks and buffers allocated for each size class. Message block headers are also allocated based on these numbers: the number of message blocks is 1.25 times the total of all data block allocations. This provides a message block for each data block, and extra blocks for duplicating messages. The optimal configuration depends on both the primary |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | memory available and the intended application. LAN Manager client increases the value of NBLK16 through NBLK512 by 40, the value of NBLK1024 by 16, the value of NBLK2048 by 8, and the value of NBLK4096 by 4.                                                                                                                                                                                                                                                   |
| <b>NQUEUE</b>  | the number of Streams queues configured. Queues are always allocated in pairs, so this number should always be even. A minimal Stream contains four queues (two for the Stream head, two for the driver). Each module pushed on a Stream requires two additional queues. A typical value is four times the value in NQUEUE. LAN Manager client increases the value of NQUEUE to 240.                                                                              |
| <b>NSTREAM</b> | the number of "Stream-head" (stdata) structures configured. One structure is needed for each Stream opened, including both Streams currently open from user processes and Streams opened linked under multiplexers. The recommended configuration value depends on the applications running on your machine. A value of 32 to 40 is usually sufficient for a computer with a single transport and moderate traffic. LAN Manager client sets this parameter to 60. |
| <b>NUMTIM</b>  | the maximum number of Streams modules that can be pushed by the Transport Library Interface (TLI). This value controls the number of data structures used to hold pushed Streams modules configuration data. LAN Manager client sets this parameter to 128. You should not modify it.                                                                                                                                                                             |
| <b>NUMTRW</b>  | the number of Transport Library Interface (TLI) read/write data structures to allocate in kernel data space. LAN Manager client sets this parameter to 128. You should not modify it.                                                                                                                                                                                                                                                                             |

The kernel parameters shown in Table 6-3 are added to *mtune* when LAN Manager client is installed.

**Table 12-3 Kernel parameters added to *mtune***

| Parameter   | Default | Minimum | Maximum |
|-------------|---------|---------|---------|
| CNCBS       | *       | 0       | 128     |
| CORMAPNCB   | *       | 0       | 128     |
| MAXSESSIONS | 15      | 0       | 254     |
| NALIAS      | 16      | 0       | 1000    |
| NASEVENT    | 40      | 0       | 1000    |
| NBIDS       | 32      | 0       | 1000    |
| NBINDX      | 80      | 0       | 1000    |
| NCALLRETRY  | 10      | 0       | 10000   |
| NCONSTABLE  | 32      | 0       | 1000    |
| NFSBUFS     | 60      | 4       | 1000    |
| NLBC        | 32      | 8       | 128     |
| NLBU        | 32      | 8       | 128     |
| NNCBS       | 32      | 0       | 256     |
| NNCB_NAMES  | 16      | 0       | 256     |
| NPTE        | 64      | 0       | 1000    |
| NRECYCLE    | 3       | 1       | 32      |
| NTIDS       | 32      | 0       | 1000    |
| NWB         | 32      | 0       | 1000    |
| SRVINUSE    | 0       | 0       | 1       |
| XITONCLOSE  | 0       | 0       | 1       |
| XNETINODE   | 64      | 1       | 1024    |

- \* The default varies depending on the number of NetBIOS drivers configured. See the description of the variables below for full details.

Use the following guidelines to modify these parameters:

- CNCBS** number of coresident NCB structures allocated. The default is 0 if only one NetBIOS driver is configured, and is set to 32 if two NetBIOS drivers are configured. If you increase NNCB on a coresident system, you should increase CNCBS to the same value.
- CORMAPNCB** structure used in mapping NCB's to NetBIOS sessions. The default is 0 if only one NetBIOS driver is configured, and is set to 32 if two NetBIOS drivers are configured. If you increase NNCB on a coresident system, you should increase CORMAPNCB to the same value.
- MAXSESSIONS** maximum number of concurrent NetBIOS sessions allowed. You may want to set the value higher to accommodate sessions for **vtp**. **MAXSESSIONS** should not exceed the number of sessions configured in the NetBIOS driver.
- For more information on configuring TCP/NetBIOS, see the "Kernel configuration" section of the chapter "TCP/IP network administration" in this guide.
- NALIAS** number of alias table entries. The value for **NALIAS** specifies the number of 12-word structures (nc alias) reserved in the kernel data segment. Each time a client accesses the server with a **net use** command, an entry in this table is used. This table tracks the mapping of the alias name to the appropriate virtual circuit. If all table space is used, the **net use** command returns an error to the user. The default is 16.
- NASEVENT** number of asynchronous event calls. The default is 64. One event call is used for each outstanding read-ahead or write-behind request. If this parameter is set too low, the following message is displayed:
- LAN Manager: low on async cells (NASEVENT)
- To correct this problem, use **id tune** to increase the size of **NASEVENT**, or decrease the size of the read-ahead and write-behind window in the *constable* file.
- NBIDS** number of bind table entries. It specifies the number of 11-word structures (bidtab) reserved in the kernel data segment. Virtual circuits using an SMB dialect other than CORE use these structures to bind an ID with a pathname for later use. The default is 30.

**NBINDX** number of BINDX structures. The default is twice the value of NFSBUFS (see the entry for NFSBUFS later in this section). It is used for tracking read-ahead requests. If this parameter is set too low, the following message is displayed:

LAN Manager: low on bindxs (NBINDX)

For more information, read the section "Modifying the client for performance: constable" earlier in this chapter.

**NCALLRETRY** maximum number of times LAN Manager client tries to establish a session if the initial connection fails. Retries, which generally succeed, occur when certain conditions on the LAN Manager server exist. The default for LAN Manager client is 10.

**NCONSTABLE** number of client table structures. The entries in this table correspond to the number of servers in the */usr/lib/lm/constable* and */usr/lib/lm/conspw* files. When you add servers to these files, increase the value of NCONSTABLE accordingly. The default for LAN Manager client is 32.

**NFSBUFS** maximum number of network buffers. This parameter determines the total number of network buffers available. Reducing this number increases the amount of memory available for user programs. A 6-word structure (*nfsbuf*) is reserved in the kernel data segment for each network buffer specified in *stune*. This structure contains a pointer to the actual buffer space that is reserved when LAN Manager client is started. The default for LAN Manager client is 60.

**NLBC** maximum number of streams that can be configured by the connection-oriented loopback driver. The default is 32.

**NLBU** maximum number of streams that can be configured by the connectionless loopback driver. The default is 32.

**NNCB** number of NCBs. This number should not be configured higher than the NNCB value for the NetBIOS driver. Refer to your NetBIOS driver documentation for information on how the driver sets this value. The default for LAN Manager client is 32.

For more information on configuring TCP/NetBIOS, see the "Kernel configuration" section of the chapter "TCP/IP network administration" in this guide.

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NNCB_NAMES</b> | length of the NCB names. This number should not be configured higher than the the number of NetBIOS names configured on the NetBIOS driver. Refer to your NetBIOS driver documentation for information on how the driver sets this value. The default for LAN Manager client is 16.<br><br>For more information on configuring TCP/NetBIOS, see the "Kernel configuration" section of the chapter "TCP/IP network administration" in this guide.                                                                                                                                                                                                                                                                 |
| <b>NPTE</b>       | number of process environment table entries. This parameter specifies the number of 11-word structures (nptab) reserved in the kernel data segment. These structures are used by the network software to track network resources. Each process using the CORE protocol dialect to talk to the server and each process using the extended protocol require an entry in this table. These entries are used when a local process first references a remote file. The entry is released when the process exits. The following message:<br><br>out of network process table space (NPTE)<br><br>appears when this parameter is set too low for the activity on the network. The default for LAN Manager client is 64. |
| <b>NRECYCLE</b>   | maximum number of dormant virtual circuits that are recycled, when the virtual circuit table is full and the client needs a new virtual circuit. The default for LAN Manager client is 3.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>NTIDS</b>      | number of tree-connect table entries. This value specifies the number of 5-word structures (tidtab) reserved in the kernel data segment. Tree connects bind a client alias with a server. The default for LAN Manager client is 32.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>NWB</b>        | number of write-behind structures. One NWB is allocated for each file written to. The following message:<br><br>LAN Manager: out of write behind table entries (NWB)<br><br>indicates that more NWBs should be allocated. The default for LAN Manager client is 32.                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>SRVINUSE</b>   | tells the operating system that the server is not configured, and the kernel data space required for the server should not be allocated. The value in <i>mtune</i> is 0. The network administrator should not modify this value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

- XITONCLOSE** allows all virtual circuits to become dormant when all files are closed. If this boolean variable is set to non-zero, virtual circuits connecting the client to server resources become dormant when all files are closed. If it is zero (0), virtual circuits remain active until all processes connected to remote resources over that circuit have exited. Leaving the circuit active means that subsequent opens have less overhead on the server. Applications that frequently open and close many files benefit from this. In contrast, if you allow circuits to become dormant, the server then has the option of recycling the circuit to serve another client on the network. The default for LAN Manager client is 0.
- XNETINODE** number of inode structures allocated for LAN Manager client.

To modify parameters using **configure(ADM)** enter **configure** at the UNIX system prompt. This interactive program prompts you for new values, which you then enter. When you exit **configure**, it updates the necessary files and relinks a new kernel.

To modify parameters using **id tune(ADM)**, enter **id tune** at the UNIX system prompt. The syntax is:

**id tune [-f] [-m] *name value***

where *name* is the name of the parameter you want to change and *value* is the new value for that parameter. For a complete description of **id tune** syntax, refer to the **id tune** manual page. After you change kernel parameters using **id tune**, you must relink the kernel using **link\_unix**. The procedure for relinking the kernel using the Link Kit is described in the *Administering Operating System Services* section of this book.

### ***TCP/IP NetBIOS driver parameters***

---

For detailed information on configuring TCP/IP NetBIOS driver parameters, see the "Kernel configuration" section of the "TCP/IP network administration" chapter in this guide.

## Chapter 13

# *Building a remote network with UUCP*

---

This chapter explains how to use UUCP to build a remote network system for your computer using a normal telephone line and a modem.

**NOTE** UUCP is not a terminal emulation program. If you want to use your modem to dial into another computer and log on, you should refer to the "Using modems" chapter of this guide and follow the instructions for adding dial-in and dial-out modems.

If you plan to do extensive file transfers between physically separated XENIX and UNIX systems, you should set up a UUCP connection.

## *What is UUCP?*

---

The UUCP package permits XENIX and UNIX systems to communicate as part of a remote network. The name UUCP is an acronym for "UNIX-to-UNIX Copy". The UUCP package consists of a group of programs that provide the following capabilities:

- remote file transfer (**uucp**)
- remote command execution (**uux**)
- mail to and from remote sites (via **mail**)

The UNIX system uses the HoneyDanBer implementation of UUCP. Used primarily over phone lines, UUCP can connect with specific remote machines on a demand or scheduled basis, and by either dialing out or allowing other machines to call in.



UUCP uses a batch method to manage communications traffic, storing (or “spooling”) requests for later execution when actual contact is made between systems. When UUCP commands are executed, work files and any data files needed are created in `/usr/spool/uucp` and its subdirectories. The program **uucico** scans these directories for the instructions contained in any work files and executes them. Although it is possible to execute commands immediately, most systems call other systems according to a daily schedule (usually during the evenings to reduce connection costs).

## *How to use this chapter*

---

This chapter describes how to build a UUCP system and covers both hardware installation and software configuration. There are also sections on routine maintenance and troubleshooting.

The following is a procedural outline of what must be done to set up your UUCP network:

1. Connect and configure a modem or direct wire.
2. Configure the UUCP software by editing the configuration files, or by using the **uinstall** utility.
3. Create login accounts for any sites that will be calling your system.
4. Test your connections with each remote site.

**NOTE** If you are planning to route mail over your UUCP system, see the chapter “Setting up electronic mail” in this guide for instructions on configuring mail traffic to work over UUCP.

The most important task of configuring UUCP is the editing of several control files that act as the database for UUCP. The next few sections describe the function of these files, and “Configuring UUCP on your system” (page 665) explains the information that these files contain. You can edit these files manually, or you can use **uinstall**, a utility that edits them for you and explains each entry. **uinstall** also includes an extensive help facility. Read “Configuring UUCP on your system” (page 665) carefully before running **uinstall** so that you understand the UUCP database.

## What you need

---

To set up your UUCP communication system, you need:

- at least one RS-232 serial line (or serial port) on your computer to use for UUCP
- the UUCP and MAIL packages extracted from your UNIX system distribution using `custom(ADM)`
- a modem, configured as described in the “Using modems” chapter of this guide
- a standard telephone jack for access to the telephone system
- a cable to connect the serial port to the modem

## UUCP commands

---

UUCP programs are divided into two categories: user programs and administrative programs. The paragraphs that follow describe the programs in each category.

### User programs

---

The user programs for basic networking are in `/usr/bin`. No special permission is needed to use these programs, although it is possible to restrict access to the devices they control. These commands are all described in the *User's Guide*.

### Administrative programs

Most of the administrative programs, control files, and scripts are in `/usr/lib/uucp`. Two exceptions are `uuinstall` and `uulog`, which are in `/etc` and `/usr/bin`, respectively.

|                |                                                                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>uucheck</b> | checks for the presence of basic networking directories, programs, and support files. It also checks the <i>Permissions</i> , <i>Systems</i> , and <i>Devices</i> files for syntax errors.                                                             |
| <b>uuclean</b> | cleans up the spool directory. It is normally executed from a shell script called <code>uudemon.clean</code> , which can be set up to be run by <code>cron</code> .                                                                                    |
| <b>uulog</b>   | displays the contents of a specified computer's log files. Log files are created for each remote computer your computer communicates with. The log files contain records of each use of <code>uucp</code> , <code>uuto</code> , and <code>uux</code> . |
| <b>uustat</b>  | displays the status of requested transfers ( <code>uucp</code> , <code>uuto</code> , or <code>uux</code> ).                                                                                                                                            |

**uutry** tests call-processing capabilities and displays debugging information. It invokes the **uucico** daemon to establish the communications link.

## **UUCP directories**

---

There are three directories associated with UUCP:

**/usr/spool/uucp** This is the working directory for UUCP. Work files, lock files, log files, and all UUCP communications traffic are stored here and in subdirectories.

**/usr/spool/uucppublic** This is the publically readable or writable target directory used for most file transfers. A tilde (~) can be used as an abbreviation for this directory in **uucp** commands (in **cs**h it must be escaped: \~).

**/usr/lib/uucp** Most of the UUCP programs are stored here, as well as the supporting database or control files. The main user programs, including **uux** and **uucp**, are found in **/usr/bin**.

The **/usr/lib/uucp** directory also contains configuration files for UUCP (distinguished by their capitalized names). The most important to understand are:

**Systems** contains information needed to establish a link to a remote computer, including the name of the connecting device associated with the remote computer, when the computer can be reached, telephone number, login sequence, and password.

**Permissions** defines the access level granted to computers when they attempt to transfer files or remotely execute commands on your computer.

**Devices** contains information concerning the port name, speed, and type of the Automatic Call Units (modems), direct links, and network devices.

## **UUCP background programs**

---

The **uucp** traffic is managed by three *daemons*, or supervisory programs, that run in the background, handling file transfers and command executions. (The daemons can also be executed manually as commands.)

**uucico** selects the device used for the link, establishes the link to the remote computer, performs the required login sequence and permission checks, transfers data and executes files, logs results, and (if requested) notifies the user by mail of transfer completions. When the local **uucico** daemon calls a remote computer, it “talks” to the **uucico** daemon on the remote computer during the session.

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>uuxqt</b>   | performs remote program execution. <b>uuxqt</b> runs after the conversation between the <b>uucico</b> programs is completed. It searches the spool directory for execute files ( <i>X,files</i> ) that were sent from a remote computer. When an <i>X,file</i> file is found, <b>uuxqt</b> opens it to get the list of data files that are required for the execution. It then checks to see if the required data files are available and accessible. <b>uuxqt</b> also verifies that it has permission to execute the requested command. |
| <b>uusched</b> | schedules the queued work in the spool directory. Before starting the <b>uucico</b> daemon, <b>uusched</b> randomizes the order in which remote computers are called.                                                                                                                                                                                                                                                                                                                                                                     |

## How UUCP works

---

When you enter a UUCP command, the program creates a work file and usually a data file for the requested transfer. The work file contains information required for transferring the file(s). The data file is a copy of the specified source file. After these files are created in the spool directory, the **uucico** daemon is started.

The **uucico** daemon attempts to establish a connection to the remote computer. First it gathers the information required for establishing a link to the remote computer from the *Systems* file. This is how **uucico** knows what type of device to use in establishing the link. Next, **uucico** searches the *Devices* file looking for the devices that match the requirements listed in the *Systems* file. After **uucico** finds an available device, it attempts to establish the link and log in on the remote computer.

When **uucico** logs in on the remote computer, the **uucico** daemon is started on the remote computer. The two **uucico** daemons then negotiate the line protocol to be used in the file transfer(s). The local **uucico** daemon then transfers the file(s) that you are sending to the remote computer. The remote **uucico** places the file in the specified pathname(s) on the remote computer. After your local computer completes the transfer(s), the remote computer may send files that are queued for your local computer. The remote computer can be denied permission to transfer these files with an entry in the *Permissions* file. (This is also affected by directory permissions.) If this is done, the remote computer must establish a link to your local computer to perform the transfers. A remote computer can also request files.

If the remote computer or the device selected to make the connection to the remote computer is unavailable, the request remains queued in the spool directory. If set up to run by **cron** each hour, **uudemon.hour** starts the **uusched** daemon. When the **uusched** daemon starts, it searches the spool directory for the remaining work files, generates the random order in which

these requests are to be processed, and then starts the transfer process (**uucico**) described in the previous paragraphs.

## A sample UUCP transaction

---

The following steps trace the execution of a **uucp** command:

1. A user on a system called *kilgore* wishes to send a copy of the file *minutes.01.10* to a remote system called *obie*. To accomplish this, the user enters the following command:

```
uucp minutes.01.10 obie\\usr/spool/uucppublic
```

Note that **obie\\!~** would also work for the destination and the exclamation point need only be escaped (preceded by a "**\**") if the **cs**h is used; the Bourne shell (**sh**) and Korn Shell (**ksh**) do not require this.

2. A work file is created in the */usr/spool/uucp/obie* directory, *C.obieNxxxx*, where *xxxx* is the job number.
3. The **uusched** daemon schedules the request for execution by **uucico**.
4. When the execution time is reached, **uucico** first checks the *Systems* file and confirms that *obie* is a recognized system and that a call is permitted at this time.
5. Using the information in the *Systems* file, **uucico** next locates the modem device and tty port associated with it as stored in the *Devices* file.
6. Using the phone number in the *Systems* file and the modem type from the *Devices* file, **uucico** uses the appropriate modem commands from the *Dialers* file (or runs a dialer program from the */usr/lib/uucp* directory) to connect to the remote system.

**Table 13-1 Example UUCP control files (sites: kilgore and obie)**

```
Systems: obie Any ACU 2400 14081234567 --ogin:-BREAK-ogin:
 nuucp ssword: mavra
```

```
Devices: ACU tty1A - 2400 dialHA24
```

```
Permissions: LOGNAME= ukilgore MACHINE= kilgore \
 READ=/usr/spool/uucppublic:/usr/kilgore \
 WRITE=/usr/spool/uucppublic:/usr/kilgore \
 REQUEST=no SENDFILES=call \
 COMMANDS=rmail:rnews:uucp
```

7. **uucico** creates a lock file (*LCK..tty1a*) to lock the serial line, and a lock file (*LCK..obie*) to lock the called system in the directory */usr/spool/uucp*.
8. **uucico** uses the login sequence and password defined in the *Systems* file to log in to *obie*, whose own **uucico** confirms that *kilgore* is recognized before beginning the actual transaction.

9. The calling system, *kilgore*, is said to be the “guest”; the called system, *obie*, is said to be the “host”. The host *uucico* checks the local *Permissions* file to confirm that the guest is authorized to transfer the file.
10. The guest (*kilgore*) transmits the file in packets that are checked for errors and retransmitted if garbled. During reception, the file is stored in a temporary file (*TM.xxxx*) in the */usr/spool/uucp/kilgore* directory on the host (*obie*). When the transfer is complete, the file is moved to the proper destination, in this case */usr/spool/uucppublic/minutes.01.10*.
11. Each machine records its side of the transaction in log files. For example, *obie* would have recorded the exchange in */usr/spool/uucp/.Log/uucp/kilgore*.
12. Unless the host system (*obie*) has requests of its own, a hangup request is sent, the connection is terminated, and the lock files are removed.

For remote command execution (via *uux*), an execute *X.file* is created in the */usr/spool/uucp* directory. The *uuxqt* daemon scans this directory for work, checks the *Permissions* file to confirm permission to execute the command, then executes it. This takes place after the modems hang up and *uucico* exits.

## Configuring UUCP on your system

To configure your UUCP system, you must connect a modem and edit a series of files that contain information about, and control the actions of, the UUCP programs.

**NOTE** To configure UUCP over a direct wire, see “Connecting two local systems using a direct wire” (page 695) for instructions.

The UUCP control files are in the */usr/lib/uucp* directory. You can modify these files with a standard text editor, or you can use the *uuninstall* utility. The descriptions found in “Detailed descriptions of UUCP configuration files” (page 669) provide details on the structure of these files so that you can create more complex configurations than the examples provided.

**NOTE** After configuring UUCP, if you have any problems initiating transactions, see the section on UUCP in the “Troubleshooting Reference” chapter of *Performance and Troubleshooting* for helpful information.

## Connecting a UUCP modem

To configure and install a modem, follow the instructions in the “Using modems” chapter of this guide and return to this section after your modem is up and running.

## Variable rate modems

Some modems can determine the connection baud rate from the carrier sent by a remote system. These modems inform the local system of the connection baud rate before issuing the Carrier Detect (CD) signal. The Hayes 2400 dialer supplied with UUCP detects different connection baud rates and informs UUCP and cu when it exits with a successful connection.

The speed fields in *Devices* and *Systems* can specify a range of baud rates for a connection. If a dialer supports baud rates from 300 to 2400 baud, enter the baud rate range in the speed field of *Devices* as follows:

**300-2400**

If a dialer or modem does not allow variable baud rates, place a single baud rate in the speed field. If a remote system supports several different speeds, place the range of baud rates in the speed field of *Systems*. If the remote system connects at a single baud rate, place that number in *Systems*. UUCP passes the highest common speed of the *Systems* and *Devices* baud rate ranges to the dialer when connecting. If the dialer connects outside of the baud range in the *Systems* file, it returns a bad baud rate error. Otherwise, it returns the baud rate of the connection.

## Editing the UUCP control files

---

This section is concerned with the configuration or control files that act as the UUCP database. A simple configuration is assumed in this section; more detailed descriptions of the UUCP files are found later in this chapter.

To configure the UUCP files, do the following:

1. Set up the *Systems* file on each machine. Use the following format:

```
sitename Any ACU baud phone# -\r\d-ogin:-\K\d-ogin:-\K\d-ogin:-\K\d-ogin:
nuucp word: password
```

where:

|                 |                                            |
|-----------------|--------------------------------------------|
| <b>sitename</b> | login name for the opposite site           |
| <b>baud</b>     | baud rate (or range) used for dial-out     |
| <b>phone#</b>   | phone number of opposite site              |
| <b>password</b> | password for UUCP account at opposite site |

**NOTE** Some lines may exceed 80 characters in length, but should still be treated as a single line. In other words, do not press (Return) when the text reaches the right side of the screen. Press (Return) only when you have finished typing in the line.

Sitenames should be no longer than seven characters and should contain no 8-bit characters (some sites reject both). In addition, they should not contain control characters or escape sequences.

The baud rate should be set to the highest common baud rate between the modems that will be used. In other words, if the modem on machine A is a 2400 baud modem, and the modem on machine B is a 1200 baud modem, then these should both be set for 1200.

Here is a set of example entries for a pair of sites, *kilgore* and *obie*, with 2400 baud modems:

**site: obie**

```
kilgore Any ACU 2400 5551212 -\r\d-ogin:-\K\d-ogin:-\K\d-ogin:-\K\d-ogin:
nuucp word: Troutster
```

**site: kilgore**

```
obie Any ACU 2400 5551212 -\r\d-ogin:-\K\d-ogin:-\K\d-ogin:-\K\d-ogin:
nuucp word: mAvraC
```

**NOTE** If you are configuring UUCP to function over a leased line, use "Direct" in place of "ACU" in the *Systems* entries. The same is true for direct wire connections.

There are many other specifications that can be included in the *Systems* file; it is discussed in more detail in "Adding entries for remote sites to the *Systems* file" (page 670).

2. Set up the *Permissions* file on each machine. Add the following text to the bottom of the file `/usr/lib/uucp/Permissions` on each machine, leaving a blank line between any text already in the file and the following new text, substituting the respective site name for *site*:

```
MACHINE=site LOGNAME=login \
COMMANDS=rmail:rnews:uucp \
READ=/usr/spool/uucppublic:/usr/tmp \
WRITE=/usr/spool/uucppublic:/usr/tmp \
SENDFILES=yes REQUEST=yes
```

Note that the **LOGNAME** must be the login name that the site uses. There are many other options that can be included in the *Permissions* file; it is discussed in more detail in "Limiting access with the *Permissions* file" (page 676).

## Creating login accounts for sites dialing-in

A dial-in site must provide a login entry for the sites that call it.

A UUCP login account is the same as an ordinary user account but it has a special login directory and login program instead of the normal user directory and shell. (Refer to *Administering Operating System Services* in this book for more information on creating login accounts).

**NOTE** "uucp" should not be used as the name of a UUCP user or login account; it is the name of the UUCP owner or administrator.



To create a UUCP login entry, follow these steps:

1. Choose a new user name and a user ID (identification number) for the UUCP login. The name can be any combination of letters and digits that is no more than eight characters long. The user ID must be an integer in the range 50 to 65535.

Make sure the name and ID are unique. A UUCP login entry must not have the same name or ID as any other login entry.

2. To create the new account, invoke **sysadmsh** and make the following selection:

Accounts ⇨ User ⇨ Create

3. Use the following information to create the account:

Login shell: */usr/lib/uucp/uucico*

Home directory: */usr/spool/uucplogins/username*

Passwords are optional, but recommended, for UUCP logins.

## ***UUCP anonymous login accounts***

---

UUCP login accounts are created with a default password expiration of 14 days. To alter this, you must use the **sysadmsh(ADM)** selection Accounts ⇨ User ⇨ Examine:Expiration to redefine this limit. For more information, see the "Administering user accounts" chapter of *Administering Operating System Services* in this guide.

**NOTE** Remember that UUCP login accounts are used by remote systems using a login script which cannot cope with a prompt for a new password. For this reason it is sensible to set up an infinite password expiration, with the password changed manually in consultation with the remote site using that UUCP login.

If you have difficulties with UUCP accounts being locked (messages like "dead account" are displayed), you can extend the number of login attempts by selecting Accounts ⇨ User ⇨ Examine:Logins. If the account was locked due to too many unsuccessful login attempts, the "Account Locked" field displays "Too many unsuccessful login attempts". You can clear this condition either by setting the maximum unsuccessful logins to a larger number (including infinite), or by selecting the Lock status option Clear all locks.

## ***Testing the UUCP connection***

---

To test your UUCP connection, follow these steps:

1. If you are configuring an outbound UUCP connection, **cu** must function in order for UUCP to work. Be sure and follow the instructions for configuring and testing the modem in the "Using modems" chapter of this guide.

2. If you are using a Hayes or compatible modem, make sure the volume on the modem is at an appropriate level. You must be able to hear the modem to carry out this test successfully.
3. Ensure that the *Systems* file has an entry for the system you intend to call, and that the *Devices* file has a matching entry for *ttynn*.
4. Start the *uutry* program by entering:  
`/usr/lib/uucp/uutry -r -x9 sitename`

**NOTE** The output from this test can be found in */tmp/sitename*. If you need to call your provider for assistance, it is a good idea to save this output.

5. Listen carefully to the modem. You should hear each digit as the number is dialed, then hear a high-pitched signal when the other modem connects, followed by silence.
6. The dialer automatically disconnects any call that it cannot complete. To break out of the shell created by *uutry*, press <Del> or <BREAK>. This returns control to the terminal while *uucico* continues to run, sending the output to a file in */tmp* with the name of the system called.
7. If the signal is not present, make certain:
  - the modem is connected to the telephone jack
  - the jack is connected to the phone system
  - the correct phone number is in the *Systems* file
8. If you do not hear the modem dial, make certain:
  - the volume switch is up
  - the modem is connected to the correct serial line and that the cable connection is tight
  - the correct tty line is in the *Devices* file
  - the modem's power is on
  - there are no *LCK..* files in */usr/spool/uucp*

## Detailed descriptions of UUCP configuration files

---

This section includes detailed information on the UUCP database files. Sites with specific needs can design files as needed rather than relying on the simple examples included in this chapter.

## Adding entries for remote sites to the *Systems* file

---

The *Systems* file (*/usr/lib/uucp/Systems*) contains the information needed by the *uucico* daemon to establish a communications link to a remote computer. Each entry in the file represents a computer that can be called by your computer.

**NOTE** If you plan to route mail traffic over UUCP, you must also configure MMDF as described in the “Setting up electronic mail” chapter of this guide.

In addition, the *Systems* file can be configured to prevent any computer that does not appear in this file from logging in on your computer. More than one entry may be present for a particular computer. The additional entries represent alternative communication paths that can be tried in sequential order.

**NOTE** If you are setting up your system as a dial-in only (passive) site that never initiates calls, you only need to add the names of the systems that will be calling you with the keyword “Never” as in this example:

```
guardian Never
```

Each entry in the *Systems* file has the following format (each field must be separated by a space):

*sitename schedule device speed phone login-script*

where:

|                     |                                                                                                                                                                                                                                                                                                                     |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>sitename</i>     | contains the node name of the remote computer.                                                                                                                                                                                                                                                                      |
| <i>schedule</i>     | is a string that indicates the day-of-week and time-of-day when the remote computer can be called.                                                                                                                                                                                                                  |
| <i>device</i>       | is the device type that should be used to establish the communications link to the remote computer.                                                                                                                                                                                                                 |
| <i>speed</i>        | indicates the transfer speed (or range) of the device used in establishing the communications link.                                                                                                                                                                                                                 |
| <i>phone</i>        | provides the phone number of the remote computer for automatic dialers. If you wish to create a portable <i>Systems</i> file that can be used at a number of sites where the dialing prefixes differ (for internal phone systems), refer to “Using Dialcodes to create a portable <i>Systems</i> file” (page 688) . |
| <i>login-script</i> | contains login information (also known as a “chat script”).                                                                                                                                                                                                                                                         |

## The *schedule* field

The *schedule* field consists of three subfields. The first, *day*, is required. The other two, *time* and *retry*, are optional. The syntax is as follows:

*day*[*time*][*retry*]

The *day* subfield can contain the following keywords:

**Su Mo Tu We** for individual days

**Th Fr Sa**

**Wk** for any weekday (**Mo Tu We Th Fr**)

**Any** for anytime

**Never** for a passive arrangement with the remote computer. If the *schedule* field is **Never**, your computer never initiates a call to the remote computer. (This field is ignored when you set up polling with **uudemon.poll2**; see "Setting up polling" (page 676) for details.) The call must be initiated by the remote computer. In other words, your computer is in a passive mode with respect to the remote computer (see discussion of *Permissions* file).

The optional *time* subfield should be a range of times in 24-hour clock format, such as 0800-1230. If no *time* is specified, any time of day is assumed to be allowed for the call. A time range that spans 0000 is permitted. For example, 0800-0600 means all times are allowed other than times between 6 AM and 8 AM.

For example, the following permits calls on Mondays, Wednesdays, and Fridays between the hours of 9 am and noon (the *schedule* field is in boldface for clarity):

```
grebe MoWeFr0900-1200 ACU D1200 14087672676 ogin:
nuucp ssword: Crested
```

You can also specify more than one set of *day* and *time* entries by separating them with commas. This is useful for more complex specifications. The following example allows calls from 5 PM to 8 AM, Monday through Thursday, and calls any time on Saturday and Sunday. This example would be an effective way to call only when phone rates are low, if immediate transfer is not critical:

```
gorgon Wk1700-0800, SaSu ACU D1200 14087672676 ogin:
nuucp ssword: DontLook
```

The optional subfield, *retry*, is available to specify the minimum time (in minutes) before a retry following a failed attempt. The subfield separator is a semicolon (;). For example, the following is interpreted as “call any time, but wait at least 9 minutes before retrying after a failure occurs”:

Any;9

By default, UUCP uses a method called exponential backoff to allow retry of failed calls. UUCP does not allow another call to go through until after the retry time has elapsed. This interval expands exponentially as the number of unsuccessful attempts increases. The *retry* field overrides the exponential backoff algorithm. If you set the retry field to 9, for example, UUCP allows another attempt to connect 9 minutes after each failure. The *retry* field cannot be set lower than 5 minutes.

UUCP does not automatically try a failed call again. You must have polling set up as described in “Setting up polling” (page 676) or manually invoke *uucico*(ADM). Any files not transferred due to a connection failure are transferred at the next successful connection to that system.

### *The device field*

The *device* field selects the device type, in most cases an ACU (Automatic Call Unit). For example, the keyword used in the following field is matched against the first field of *Devices* file entries:

```
Systems: gorgon Any ACU D1200 14087672676 ogin:
 nuucp ssword: DontLook
```

```
Devices: ACU tty2A - 1200 /usr/lib/uucp/dialHA12
```

Additional documentation on the *device* field is located in the */usr/lib/uucp/Systems* file supplied with your system.

### *The speed field*

This field can contain a letter and speed (for example, C1200, D1200) to differentiate between classes of dialers (refer to the discussion on the *Devices* file, *speed* field). Some devices can be used at any speed, so the keyword **Any** can be used. However, we recommend that you specify the actual range of speeds that can be used. (If **Any** is used in both *Systems* and *Devices* entries, 1200 is assumed.) For example, this field must intersect the *speed* field in the associated *Devices* file entry:

```
Systems: gorgon Any ACU D2400-9600 14087672676 ogin:
 nuucp ssword: DontLook
```

```
Devices: ACU tty1A - D2400-9600 /usr/lib/uucp/dialHA9600
```

If information is not required for this field, use a hyphen (-) as a place holder for the field.

## The phone field

This field provides the phone number used for the modem dialer. The phone number is made up of an optional alphabetic abbreviation and a numeric part. If an abbreviation is used, it must be one that is listed in the *Dialcodes* file. (See “Using Dialcodes to create a portable Systems file” (page 688) for details.) For example:

```
Systems: gorgon Any ACU D1200 CA2676 ogin:
 nuucp ssword: DontLook
```

```
Dialcodes: CA 9=408767
```

In this string, an equal sign (=) tells the ACU to wait for a secondary dial tone before dialing the remaining digits. A dash in the string (–) instructs the ACU to pause 2 seconds before dialing the next digit.

**NOTE** Most dialers treat the equal sign as a pause, so you may need to use more than one.

Do not use the comma (,) from the Hayes command set in a *Systems* file entry when you wish to indicate a pause. Use hyphens instead.

If your computer is connected to a LAN switch or port selector, you can access other computers that are connected to that switch. The *Systems* file entries for these computers do not have a phone number in the *phone* field. Instead, this field contains the token that must be passed on to the switch so it knows which computer your computer wishes to communicate with. (This is usually just the system name.) The associated *Devices* file entry should have a “\D” at the end of the entry to prevent translation using the *Dialcodes* entry.

## The login-script field

The login-script opens communications between modems, and also recognizes and sends proper login and password sequences. The script is given as a series of space-separated fields and subfields of the following format:

*expect send*

where *expect* is the string that is received, and *send* is the string that is sent when the *expect* string is received.

The *expect* field can be made up of subfields of the following form:

*expect[-subsend-subexpect] ...*

where the *subsend* is sent if the prior *expect* is not successfully read and the *subexpect* following the *subsend* is the next expected string. To make this distinction clear: the send-expect sequence sends a string if the expect string is received; the subsend-subexpect sends only if the prior expect string is not received within 10 seconds.

For example, with "login:-login:", the UUCP program expects "login:". If a "login:" is received, it goes on to the next field. If it does not get "login:", it sends nothing followed by a carriage return, then looks for "login:" again. If no characters are initially expected from the remote computer, the null string ("") should be used in the first *expect* field. Note that all *send* fields are sent followed by a carriage return unless the *send* string is terminated with a "\c".

If an *expect* string starts with a dash, it is interpreted as a null *expect* string followed by a *subsend* string. For example, "--login:" sends a carriage return and then expects a "login:".

The *expect* string need not be complete; only the trailing characters must be specified, as in "ogin:". This avoids difficulties with login strings that use an uppercase letter as in "Login:" or "Password:", and also difficulties when the line is shared by dial-in and dial-out.

## Creating login scripts

This section explains in greater detail how to create a login (chat) script.

Consider the following sample *Systems* file entry:

```
terps Any ACU 1200 18005211980 "" \r ogin:-BREAK-ogin:
uucpx word: ichore
```

This is how this script would work during connection:

1. Nothing is expected initially.
2. A carriage return is sent and the script waits for the prompt "ogin:" (login:).
3. If it does not receive "ogin:", send a **BREAK** signal.
4. When "ogin:" is finally received, send the login name *uucpx*.
5. When the prompt "word:" (for Password:) is received, send the password *ichore*.

Login (chat) scripts often require some experimentation. There are cases that require one or more **BREAK** sequences before presenting a login (this is often true with variable speed modems). If you cannot obtain the necessary login sequence from the system administrator for a given site, it is a good idea to connect with the site manually. You can accomplish this using *cu* and find out what must be sent to generate a login prompt. (You can also connect with a system using a *uutry* for debugging; see the "Troubleshooting reference" chapter of *Performance and Troubleshooting* for details. There are several escape characters that cause specific actions when sent during the login sequence, some of which correspond to keystrokes; these should be included in the script where necessary. See Table 13-2.

**Table 13-2 Login (Chat) script escape sequences**

| Character    | Description                                                                                                                                                             |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>\N</b>    | sends a null character (ASCII NULL).                                                                                                                                    |
| <b>\b</b>    | sends or expects a backspace character.                                                                                                                                 |
| <b>\c</b>    | if at the end of a string, suppresses the carriage return that is normally sent. Ignored otherwise.                                                                     |
| <b>\d</b>    | delays two seconds before sending or reading more characters.                                                                                                           |
| <b>\p</b>    | pauses for approximately ¼ to ½ second.                                                                                                                                 |
| <b>\E</b>    | starts echo checking. (After this sequence is used, whenever a character is transmitted, the system waits for the character to be received before doing anything else.) |
| <b>\e</b>    | turns echo check off.                                                                                                                                                   |
| <b>\n</b>    | sends or expects a newline character.                                                                                                                                   |
| <b>\r</b>    | sends or expects a carriage-return.                                                                                                                                     |
| <b>\s</b>    | sends or expects a space character.                                                                                                                                     |
| <b>\t</b>    | sends or expects a tab character.                                                                                                                                       |
| <b>\\</b>    | sends or expects a “\” character.                                                                                                                                       |
| <b>EOT</b>   | sends EOT (end of transmission or (Ctrl)d)                                                                                                                              |
| <b>BREAK</b> | sends a BREAK signal.                                                                                                                                                   |
| <b>\K</b>    | same as BREAK.                                                                                                                                                          |
| <b>\ddd</b>  | collapses the octal digits ( <i>ddd</i> ) into a single character whose value is the ASCII character represented by that number (for example: \007).                    |
| <b>""</b>    | expects a null string.                                                                                                                                                  |



## Setting up polling

---

Use **uudemon.poll2** to set up polling. To run **uudemon.poll2**, you need an entry for the daily daemon and an entry for the hourly daemon in the */usr/spool/cron/crontabs/uucp* file as follows:

```
0 0 * * * uudemon.poll2 -d
0 * * * * uudemon.poll2
```

The **-d** flag refers to the daily daemon. The hourly daemon has no flags. The above example has the daemon run at midnight. You can change the time the daemon runs by altering the second field using a 24-hour clock.

To establish the hours and days that **uudemon.poll2** runs, you create two files: */usr/lib/uucp/Poll.hour* and */usr/lib/uucp/Poll.day*. These files contain the systems to be polled and the times and days they are polled.

A sample *Poll.hour* file follows:

```
hanna 12 1 3
raven 2 6 10w
```

If the hour is followed by a "w", **uudemon.poll2** calls the site only if there is work to be done.

A sample *Poll.day* file follows:

```
hanna 1 3 6
raven 1 2 3 4 5
```

The days of the week are integers where Sunday is 0.

## Limiting access with the Permissions file

---

If other machines will be dialing into your system, the *Permissions* file (*/usr/lib/uucp/Permissions*) specifies the permissions that remote computers have with respect to login, file access, and command execution. There are options that restrict the remote computer's ability to request files and its ability to receive files queued by the local site. Other options specify the commands that a remote site can execute on the local computer.

### Structuring Permissions file entries

Each entry is a logical line with physical lines terminated by a "\ " to indicate continuation. Entries are made up of options delimited by spaces. Each option is a name-value pair in the following format:

**name=value**

**NOTE** No spaces are allowed within an option assignment. This means that any continuations in an option assignment cannot have spaces before the "\ " or at the start of the next line.

Comment lines begin with a number sign (#) and they occupy the entire line up to a newline character. Blank lines are ignored (even within multi-line entries).

There are two types of *Permissions* file entries:

- LOGNAME** specifies the permissions that take effect when a remote computer calls your computer.
- MACHINE** specifies permissions that take effect when your computer calls a remote computer.

In this way it is possible not only to define permissions for sites calling your system, but permissions for when your site calls other machines.

### *Permissions file restrictions*

When using the *Permissions* file to restrict the level of access granted to remote computers:

- A machine cannot have more than one **LOGNAME** entry.
- Any site that is called whose name does not appear in a **MACHINE** entry, has the following default permissions or restrictions:
  - Only local send and receive requests are executed.
  - The remote computer can send files to your computer's */usr/spool/uucppublic* directory.
  - The commands sent by the remote computer for execution on your computer must be one of the default commands, usually **rmail**.

**NOTE** **LOGNAME** and **MACHINE** are often combined for convenience, but they function independently. For example, if a remote system logs in as *nuucp*, *uucico* will read the first entry containing **LOGNAME=nuucp** without regard to the **MACHINE** name.

### *Permissions options*

This section describes each option, specifies how they are used, and lists their default values.

#### **REQUEST**

specifies whether the remote computer can request to set up file transfers from your computer. When a remote computer requests to receive a file, this request can be granted or denied. The following string specifies that the remote computer can request to transfer files from your computer:

**REQUEST=yes**

The following string specifies that the remote computer cannot request to receive files from your computer:

```
REQUEST=no
```

The no value is the default value. It is used if the REQUEST option is not specified. The REQUEST option can appear in either a LOGNAME (remote calls you) entry or a MACHINE (you call remote) entry.

#### SENDFILES

specifies whether your computer can send the work queued for the remote computer. When a remote computer calls your computer and completes its work, it may ask if your computer has work queued for it.

The following string specifies that your computer can send the work that is queued for the remote computer as long as the remote computer is logged in as one of the names in the LOGNAME option:

```
SENDFILES=yes
```

This string is mandatory if your computer is in a passive mode with respect to the remote computer.

The following string specifies that files queued in your computer be sent only when your computer calls the remote computer:

```
SENDFILES=call
```

The call value is the default for the SENDFILES option. This option is only significant in LOGNAME entries because MACHINE entries apply when calls are made out to remote computers. If this option is used with a MACHINE entry, it is ignored.

#### READ and WRITE

specify the various parts of the filesystem that uucico can read from or write to. The READ and WRITE options can be used with either MACHINE or LOGNAME entries.

The default for both the READ and WRITE options is the *uucppublic* directory as shown in the following strings:

```
READ=/usr/spool/uucppublic
WRITE=/usr/spool/uucppublic
```

The following strings specify permission to access any file that can be read or written by UUCP.

```
READ=
WRITE=
```

The value of these entries is a colon-separated list of pathnames. The READ option is for requesting files, and the WRITE option for depositing files. One of the values must be the prefix of any full pathname of a file coming in or going out.

**NOTE** **READ** and **WRITE** options do not affect the actual permissions of a file or directory. For example, a directory with permissions of **700** only permits the owner to access it, and cannot be read or written by the UUCP user, no matter what access options are defined in the *Permissions* file. In addition to the proper **READ** and **WRITE** options, the paths must grant appropriate permissions to the UUCP user.

To grant permission to deposit files in */usr/tmp* as well as the public directory, the following values would be used with the **WRITE** option:

```
WRITE=/usr/spool/uucppublic:/usr/tmp
```

It should be pointed out that if the **READ** and **WRITE** options are used, all pathnames must be specified because the pathnames are not added to the default list. For instance, if the */usr/news* pathname was the only one specified in a **WRITE** option, permission to deposit files in the public directory would be denied.

You should be careful with which directories you make accessible for reading and writing by remote systems. For example, you probably do not want remote computers to be able to write over your */etc/passwd* file so */etc* should not be open to writes.

#### **NOREAD** and **NOWRITE**

specify exceptions to the **READ** and **WRITE** options or defaults. The following strings would permit reading any file except those in the */etc* directory (and its subdirectories—remember, these are prefixes) and writing only to the default */usr/spool/uucppublic* directory:

```
READ=/
WRITE=/usr/spool/uucppublic
NOREAD=/etc
NOWRITE=/etc
```

**NOWRITE** works in the same manner as the **NOREAD** option. The **NOREAD** and **NOWRITE** options can be used in both **LOGNAME** and **MACHINE** entries.

#### **CALLBACK**

specifies in **LOGNAME** entries that no transaction takes place until the calling system is called back. There are two examples of when you would use **CALLBACK**. From a security standpoint, if you call back a machine you can be sure it is the machine it says it is. If you are doing long data transmissions, you can choose the machine that is billed for the longer call.

The following string specifies that your computer must call the remote computer back before any file transfers take place:

```
CALLBACK=yes
```

The default for the **CALLBACK** option is:

```
CALLBACK=no
```

The **CALLBACK** option is rarely used. If two sites have this option set for each other, a conversation never gets started.

## COMMANDS

specifies the commands in **MACHINE** entries that a remote computer can execute on your computer. This affects the security of your system; use it with extreme care.

The **uux** program generates remote execution requests and queues them to be transferred to the remote computer. Files and a command are sent to the target computer for remote execution. Note that **COMMANDS** is not used in a **LOGNAME** entry; **COMMANDS** in **MACHINE** entries define command permissions whether you call the remote system or it calls you.

The default command that a remote computer can execute on your computer is:

```
COMMANDS=rmail
```

If a command string is used in a **MACHINE** entry, the default commands are overridden. For instance, the following entry overrides the **COMMAND** default so that the computers *owl*, *raven*, *hawk*, and *dove* can now execute **rmail**, **rnews**, and **lp** on your computer:

```
MACHINE=owl:raven:hawk:dove \
COMMANDS=rmail:rnews:lp
```

Full pathnames of commands can also be used. For example, the following command specifies that command **rmail** uses the default path:

```
COMMANDS=rmail:/usr/lbin/rnews:/usr/bin/lp
```

The default paths for your computer are **/bin**, **/usr/bin**, and **/usr/lbin**. When the remote machine specifies **rnews** or **/usr/lbin/rnews** for the command to be executed, **/usr/lbin/rnews** is executed regardless of the default path. Likewise, **/usr/bin/lp** is the **lp** command that is executed.

Including the **ALL** value in the list means that any command from the remote computer specified in the entry is executed. If you use this value, you give the remote computer full access to your computer. So, be careful; this allows far more access than normal users have.

The following string illustrates two points:

```
COMMANDS=/usr/local/bin/lc:ALL:/usr/bin/lp
```

- The **ALL** value can appear anywhere in the string; and the pathnames specified for **lc** and **lp** are used (instead of the default) if the requested command does not contain the full pathnames for **lc** or **lp**.

- The **VALIDATE** option should be used with the **COMMANDS** option whenever potentially dangerous commands like **cat** and **uucp** are specified with the **COMMANDS** option. Any command that reads or writes files is potentially dangerous to local security when executed by the UUCP remote execution daemon (**uuxqt**).

## VALIDATE

is used in conjunction with the **COMMANDS** option in **LOGNAME** entries when specifying commands that are potentially dangerous to your computer's security. It provides a certain degree of verification of the caller's identity. The use of the **VALIDATE** option requires that privileged computers have a unique login or password for UUCP transactions. An important aspect of this validation is that the login or password associated with this entry be protected. If an outsider gets that information, that particular **VALIDATE** option can no longer be considered secure. (**VALIDATE** is merely an added level of security to the **COMMANDS** option, though it is a more secure way to open command access than **ALL**.)

Careful consideration should be given to providing a remote computer with a privileged login and password for UUCP transactions. Giving a remote computer a special login and password with file access and remote execution capability is like giving anyone on that computer a normal login and password on your computer. Therefore, if you cannot trust someone on the remote computer, do not provide that computer with a privileged login and password.

The following **LOGNAME** entry specifies that if one of the remote computers that claims to be *eagle*, *owl*, or *hawk* logs in on your computer, it must have used the login *uucpfriend*.

```
LOGNAME=uucpfriend VALIDATE=eagle:owl:hawk
```

As can be seen, if an outsider gets the *uucpfriend* login or password, masquerading is trivial.

**VALIDATE** increases security by linking the **MACHINE** entry (and **COMMANDS** option) with a **LOGNAME** entry associated with a privileged login. This link is needed because the execution daemon is not running while the remote machine is logged in. In fact, it is an asynchronous process with no knowledge of what machine sent the execution request. Therefore, the real question is how does your system know where the execution files came from?

Each remote computer has its own *spool* directory on your computer. These spool directories have write permission given only to UUCP programs. The execution files from the remote computer are put in its spool directory after being transferred to your computer. When the **uuxqt** daemon runs, it can use the spool directory name to find the **MACHINE** entry in the *Permissions* file and get the **COMMANDS**

list. If the computer name does not appear in the *Permissions* file, the default list is used.

The following example shows the relationship between the **MACHINE** and **LOGNAME** entries:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
COMMANDS=rmail:/usr/local/bin/lc \
READ=/ WRITE=/

LOGNAME=uucpz VALIDATE=eagle:owl:hawk \
REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

The **COMMANDS** option line shows that remote mail and */usr/local/bin/lc* can be executed by remote users.

In the **MACHINE** entry, you must make the assumption that when you want to call one of the computers listed, you are really calling *eagle*, *owl*, or *hawk*. Any files put into one of the *eagle*, *owl*, or *hawk* spool directories is put there by one of those computers. If a remote computer logs in and says that it is one of these three computers, its execution files are also put in the privileged spool directory. You should validate that the computer has the privileged login *uucpz*.

#### Entries for OTHER systems

You may want to specify different option values for machines or logins that are not mentioned in specific **MACHINE** or **LOGNAME** entries. This may occur when there are many computers calling in that have the same set of permissions. The special name **OTHER** for the computer name can be used in a **MACHINE** or **LOGNAME** entry as follows:

```
MACHINE=OTHER \
COMMANDS=rmail:/usr/local/bin/lc

LOGNAME=OTHER \
REQUEST=yes SENDFILES=yes \
READ=/usr/spool/uucppublic \
WRITE=/usr/spool/uucppublic
```

All options that can be set for specific machines or logins can be used with the **OTHER** value, although the use of the **VALIDATE** option makes little sense.

#### Combining MACHINE and LOGNAME entries

It is possible to combine **MACHINE** and **LOGNAME** entries into a single entry where the common options are the same. For example, the following two entries share the same **REQUEST**, **READ**, and **WRITE** options:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
 READ=/ WRITE=/

LOGNAME=uucpz REQUEST=yes SENDFILES=yes \
 READ=/ WRITE=/
```

These two entries can be merged as follows:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
LOGNAME=uucpz SENDFILES=yes \
 READ=/ WRITE=/
```

**NOTE** The following command provides a useful interpretation of the *Permissions* file:

```
/usr/lib/uucp/uucheck -v | more
```

## Adding dial-out entries to the *Devices* file

The *Devices* file (*/usr/lib/uucp/Devices*) contains information for all the devices that can be used to establish a link to a remote computer. Devices are Automatic Call Units, direct links, or network connections. This file works closely with the *Dialers*, *Systems*, and *Dialcodes* files. Before you make changes in any of these files, you should be familiar with them all. A change to an entry in one file may require a change to a related entry in another file.

Each entry in the *Devices* file has the following format:

*type ttyline dialerline speed dialer-token*

where:

- type* contains one of two keywords (**Direct** or **ACU**), the name of a Local Area Network switch, or a system name.
- ttyline* contains the device name of the port associated with the *Devices* entry. For example, if the automatic dial modem for a particular entry was attached to the */dev/tty1A* line, the name entered in this field would be "tty1A".
- dialerline* is useful only for 801 type dialers, which do not contain a modem and must use an additional line. Unless you have an 801 dialer, simply enter a hyphen (-) as a placeholder.
- speed* is the speed or speed range of the device. Can also contain an indicator for distinguishing different dialer classes.
- dialer-token* contains pairs of dialers and tokens, each representing a dialer and an argument to be passed to it. The *dialer* portion can be the name of a binary dialer program, a *Dialers* file entry, or **Direct** for a direct link device.



## The *type* field

This field usually contains one of two keywords (**Direct** or **ACU**) the name of a Local Area Network switch, or a system name.

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Direct</b>    | indicates a direct link to another computer or a switch for cu connections.                                                                                                                                                                                                                                                                                                                                                                       |
| <b>ACU</b>       | indicates that the link to a remote computer is made through an Automatic Call Unit. This modem can be connected either directly to your computer or indirectly through a Local Area Network (LAN) switch.                                                                                                                                                                                                                                        |
| <b>LANswitch</b> | can be replaced by the name of a LAN switch. <b>micom</b> and <b>develcon</b> are supplied with caller scripts in the <i>Dialers</i> file. You can add your own LAN switch entries to the <i>Dialers</i> file. If you are adding a network using TCP/IP you would use the special dialer types <b>TCP</b> , <b>TLI</b> , or <b>TLIS</b> . For more information about configuring UUCP over TCP/IP, see "Configuring UUCP over TCP/IP" (page 689). |
| <b>sysname</b>   | indicates a direct link to a particular computer. ( <i>sysname</i> is replaced by the name of the computer.) This means that the line associated with this <i>Devices</i> entry is for a particular computer in the <i>Systems</i> file.                                                                                                                                                                                                          |

For example, the keyword **gorgon** used in the *type* field of the *Devices* file is matched against the third field of the *Systems* file entry:

*Devices:*   **gorgon** ttyla - 1200 direct

*Systems:*   gorgon Any **gorgon** 1200 - ogin: nuucp ssword: DontLook

You can designate a protocol to use for a device within this field. For more information, see "Defining a communications protocol" (page 688).

**NOTE** Although the use of a system name adds clarity, only defined tokens can be used.

## The speed field

In most cases, this is simply the speed of the device, if the keyword **ACU** or **Direct** is used in the *type* field. However, *speed* can contain a letter and a speed (for example, C1200, D1200) to differentiate between classes of dialers (Centrex or Dimension PBX). This is necessary because many larger offices may have more than one type of telephone network: one network may be dedicated to serving only internal office communications, while another handles the external communications. It is necessary to distinguish which lines are used for internal communications and which are used for external communications. The keyword used in the *speed* field of the *Devices* file is matched against the fourth field of the *Systems* file entries, for example:

*Devices:* ACU tty1A - **D1200** hayes1200

*Systems:* gorgon Any ACU **D1200** 3251 ogin: nuucp ssword: DontLook

Some devices can be used at any speed, so the keyword **Any** can be used in the *speed* field. If **Any** is used, the line matches any speed requested in a *Systems* file entry. If this field is **Any** and the *Systems* file *speed* field is **Any**, the speed defaults to 1200 bps. If a device can be used at a range of speeds, then the speed field can specify this range (for example, 1200-9600 or D1200-9600). This is preferable to the use of **Any**.

## The dialer-token field

This field has the following format:

*dialer* [ *token dialer token* ... ]

For a direct line, this field contains simply the word **direct**, and no token is required.

For a simple connection to a dialer, this field contains the name of the dialer, and the token is omitted; by default it is taken from the phone number field of the *Systems* file entry.

For a dialer or a network dataswitch, this field contains the name of an entry found in the *Dialers* file (**develcon** and **micom** are examples of network data switches). Other dialer types are supported by binaries instead of *Dialers* entries. (Support for 801-type dialers is provided through use of separate lines for data and the dialer. See the *Devices* file for details.) UUCP recognizes a dialer as a binary if the name begins with a **" / "** or if there is an executable file by that name in */usr/lib/uucp*.

For more information on *Dialers* entries and binaries, see "Dialing out from your computer" (page 835).

## Structuring dialer-token entries

The *dialer-token* can be structured four different ways, depending on the device associated with the entry:

- Simple modem connection

If an automatic dialing modem is connected directly to a port on your computer, the *dialer-token* field of the associated *Devices* file entry only has one pair. This pair would normally be the name of the modem. This name matches the particular *Devices* file entry with an entry in the *Dialers* file. Therefore, the *dialer* field must match the first field of the following *Dialers* file entry:

*Devices:* ACU tty1A - 1200 **ventel**

*Dialers:* **ventel** =&-% " " \r\p\r\c \$ <K\T%#\r>\c ONLINE!

Notice that only the *dialer* portion (**ventel**) is present in the *dialer-token* field of the *Devices* file entry. This means that the *token* to be passed on to the dialer (in this case the phone number) is taken from the *Phone* field of a *Systems* file entry. ("D" is implied; see Modems used with a local network switch (page 687) .) Backslash sequences are described later.

- Direct links

If a direct-link is established to a particular computer, the *dialer-token* field of the associated entry contains the keyword **direct**. This is true for both types of direct link entries, **direct** and *sysname* (refer to discussion on the *type* field).

- Local network switches

If a computer that you wish to communicate with is on the same local network switch as your computer, your computer must first access the switch and the switch can then make the connection to the other computer. In this type of entry, there is only one pair. The *dialer* portion matches a *Dialers* file entry, as shown in the following example:

*Devices:* develcon tty13 - 1200 develcon \D

*Dialers:* develcon "" "" \pr\ps\c est:\007 \E\D\e \007

*Systems:* obie develcon ACU 1200 obie --ogin:-BREAK-ogin:  
nuucp ssword: mavra

As shown, the *token* portion is "\D", which indicates that it is retrieved from the *Systems* file without translation. The *Systems* file entry for this particular computer contains the token in the *phone* field; this is normally reserved for the phone number of the computer (refer to *Systems* file, *phone* field). The "\D" ensures that the contents of the *phone* field is not interpreted as a valid entry in the *Dialcodes* file.

- Modems used with a local network switch

If an automatic dialing modem is connected to a switch, your computer must first access the switch and the switch makes the connection to the automatic dialing modem. This type of entry requires two *dialer-token* pairs. The following *dialer* portion of each pair (fifth and seventh fields of entry) are used to match entries in the *Dialers* file:

*Devices:* ACU tty14 - 1200 **develcon vent ventel**

*Dialers:* **develcon** "" "" \pr\ps\c est:\007 \E\D\e \007  
ventel =&% "" \r\p\r\c \$ <K\T%%\r>\c ONLINE!

In the first pair, **develcon** is the switch and **vent** is the token that is passed to the **develcon** switch to tell it which device to connect to your computer. This token would be unique for each LAN switch because each switch can be set up differently. Once the modem is connected, the second pair is accessed, where **ventel** is the dialer and the token is retrieved from the *Systems* file.

The following are two escape characters that can appear in the *dialer-token* field:

- \T indicates that the *Phone* field should be translated at this stage, using the *Dialcodes* file. This escape character is normally placed in the *Dialers* file for each caller script associated with an automatic dial modem (penril, ventel, and so on). The translation does not take place until the caller script is accessed.
- \D indicates that the *Phone* field should not be translated using the *Dialcodes* file. If no escape character is specified at the end of a *Devices* entry, "\D" is assumed by default when a *Dialers* script is to be used (which can itself contain a "\T" to translate the number). "\T" is assumed if a built-in or dialer binary is to be used (because there is then no later opportunity to translate the number).

## Special UUCP configuration options

---

This section contains several options that are used for special circumstances and can be ignored in most cases.

## Defining a communications protocol

---

You can define the protocol to use with each device. In most cases it is not needed since you can use the default or define the protocol with the particular system you are calling (For more information on defining the protocol, see the *Systems* file "type" field.) To specify the protocol, use the form *type,protocol* (for example, ACU,g). Table 13-3 lists the available protocols.

**Table 13-3 UUCP communications protocols**

| Protocol | Description                                                                                                                                                     |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| g        | standard UUCP protocol for connection over serial lines and modems. Uses error correction.                                                                      |
| e        | protocol for 8-bit error-free links (example: TCP, TLI, TLIS). No error correction.                                                                             |
| t        | protocol for 8-bit error-free links (example: TCP, TLI, TLIS). Checks received file size. This protocol is provided for compatibility with BSD-derived systems. |
| f        | protocol for 7-bit only error-free links (for example, some X.25 PADs). Does a checksum on the entire file.                                                     |
| x        | protocol for 8-bit X.25 error-free links. Does not work on some X.25 packet switched networks (see t protocol).                                                 |

## Adding dialers to the Dialers file

---

The *Dialers* file (*/usr/lib/uucp/Dialers*) specifies the initial conversation that must take place on a line before it can be made available for transferring data. New entries can be added to this file if your modem does not appear in the file. See "Using dialer programs" (page 835) for more information.

## Using Dialcodes to create a portable Systems file

---

The *Dialcodes* file (*/usr/lib/uucp/Dialcodes*) contains the dial-code abbreviations that can be used in the "Phone" field of the *Systems* file. This feature is intended primarily for those who wish to create a standard *Systems* file for distribution among several sites that have different phone systems and area codes. As such, the *Dialcodes* file is probably not necessary for most sites. See the *dialcodes(F)* manual page for more information.

## Creating alternate control files with Sysfiles

---

The */usr/lib/uucp/Sysfiles* file lets you assign different files to be used by *uucp* and *cu* as *Systems*, *Devices*, and *Dialers* files. This feature is useful for cases where UUCP and *cu* require different dialers. See the *sysfiles(F)* manual page for more information.

## Changing uucico packet parameters

---

An added feature is the ability to change two specialized parameters contained in the `uucico` program without having to recompile the source. These parameters are:

- `windows` specifies the size of window that the sliding-window protocol should use.
- `pktime` is the number of seconds `uucico` should wait before giving up and re-transmitting the packet being sent.

See the `uucico(ADM)` manual page for more information.

**NOTE** Changing the `uucico windows` parameters can cause compatibility problems. For example, it is best not to modify the default values if you plan on connecting to UUNET.

## Preventing unknown sites from logging in

---

The script `remote.unknown` is executed when a site whose name does not appear in your `Systems` file dials into your system. It logs the conversation attempt and fails to make a connection. If you wish to allow such “unknown” systems to log in to your system, you can change the permissions of this file so it cannot execute and your system accepts any communication requests. To do so, enter the following commands while logged in as `root`:

```
cd /usr/lib/uucp
chmod 000 remote.unknown
```

## Configuring UUCP over TCP/IP

---

There are several reasons you may wish to configure UUCP over TCP/IP. Some sites do not provide `ftpd` or `rshd` servers, and/or their respective clients, `ftp` and `rcp`. Thus transferring files across a TCP/IP network is not an option with these systems. Additionally, some versions of TCP/IP do not provide the Simple Mail Transfer Protocol, SMTP, for mail transfer. In both of these cases, setting up UUCP to use TCP/IP may be an option to allow both file transfer and exchange of mail between such systems.

In the following paragraphs, the word “client” refers to a system which executes the `uucp` or `uux` command, while the term “server” or “listener” refers to a system that responds to a request from a client system.

There are two approaches to configuring UUCP over TCP/IP:

#### TCP socket interface

The server system uses the `inetd` superserver to listen for incoming `uucp` requests on TCP port 540. When receiving a request from a client `uucico` process, the server system forks the `uucpd` daemon, which logs in the client with the shell `uucico`. The two `uucico` processes can now transfer information similar to the way they would in a standard serial line configuration.

#### TLI (Transport Layer Interface)

The server system uses a process called `listen` to wait for requests from a predefined TCP port and then when receiving a request, forks a `uucico` process directly, bypassing the standard UUCP login sequence.

**NOTE** When connecting two SCO systems, the TCP socket interface is the preferred method. When connecting an SCO system to another vendor's TCP/IP product, consult the vendor's documentation to determine which method is supported and what procedure to use at that end. If the socket interface is supported, use it for best results.

## *Configuring UUCP over TCP/IP with the TCP socket interface*

The following steps demonstrate how to configure UUCP over TCP/IP using the TCP socket interface between the systems *colossus* and *guardian*.

**NOTE** Before proceeding, verify that TCP/IP is installed and running properly.

1. On both systems, verify that the file `/etc/inetd.conf` has the following line, and that it is uncommented (with no “#” character at the beginning of the line):

```
uucp stream tcp nowait NOLUID /etc/uucpd uucpd
```

**NOTE** If your system is running SCO TCP/IP Release 1.1.3, replace “NOLUID” with “root” in the above line.

Should the program `/etc/uucpd` not exist, execute the following commands as *root*:

```
In /usr/lib/uucp/uucpd /etc/uucpd
chmod 755 /etc/uucpd
```

2. Verify that both systems have the following line in the */etc/services* file:

```
uucp 540/tcp uucpd # uucp daemon
```

3. Verify the configuration of the *nuucp* account on each machine. Make sure that the shell of this account is */usr/lib/uucp/uucico*, and that the account has a password on both machines. You will probably have to set a password for the *nuucp* account. Use the *sysadmsh* selection Accounts ⇨ User ⇨ Examine to confirm the account information and set a password.

4. Add the following line to the file */usr/lib/uucp/Systems* on *colossus*:

```
guardian Any TCP,e Any - ogin: nuucp word: password
```

Where *password* is the password of the *nuucp* account on *guardian*.

5. Add the following line to the file */usr/lib/uucp/Systems* on *guardian*:

```
colossus Any TCP,e Any - ogin: nuucp word: password
```

Where *password* is the password of the *nuucp* account on *colossus*.

6. Verify that the */usr/lib/uucp/Permissions* file on *colossus* has the following entry:

```
MACHINE=guardian LOGNAME=nuucp \
COMMANDS=rmail:rnews:uucp \
READ=/usr/spool/uucppublic:/usr/tmp \
WRITE=/usr/spool/uucppublic:/usr/tmp \
SENDFILES=yes REQUEST=yes
```

7. Verify that the */usr/lib/uucp/Permissions* file on *guardian* has the following entry:

```
MACHINE=colossus LOGNAME=nuucp \
COMMANDS=rmail:rnews:uucp \
READ=/usr/spool/uucppublic:/usr/tmp \
WRITE=/usr/spool/uucppublic:/usr/tmp \
SENDFILES=yes REQUEST=yes
```

8. Add the following line to */usr/lib/uucp/Devices* on both machines:

```
TCP TCP,e - Any TCP 540
```

9. Verify that both systems are in their partner's */etc/hosts* file, or resolvable by a nameserver.
10. If you needed to change anything in step 1, shutdown and reboot the systems where you made the changes. UUCP should then work as expected between the two systems. Each system should be able to issue requests for the opposite system.



**NOTE** Note that steps 3 through 7 are similar to the steps used in configuring UUCP between systems over a serial line. Also, the user names and sample *Permissions* files shown here are only examples and may be changed to suit the security needs of a particular site.

## Configuring UUCP over TCP/IP with TLI

The following steps demonstrate how to configure UUCP over TCP/IP using TLI between the systems *colossus* and *guardian*. The example is a configuration where the machine *colossus* initiates all UUCP requests, while the machine *guardian* listens and responds to all requests. If you want both machines to be able to use the `uucp` command to initiate a UUCP transaction, repeat this procedure twice, exchanging machine names where appropriate.

**NOTE** Before proceeding, verify that TCP/IP is installed and running properly.

1. Pick a TCP port on which *guardian* will listen for incoming UUCP requests. The following example uses port 698. The port chosen should not already appear in the `/etc/services` file, and should not be port 540.
2. Determine the address on which *guardian* will listen for requests. This number is the representation of a `sockaddr_in` structure as defined in the header file `/usr/include/sys/netinet/in.h`. This number is a 16 byte number and needs to be constructed both in hexadecimal and in octal. Use of the `bc(C)` command is helpful here. To construct the number in hexadecimal, take the following steps:
  - A. The first byte represents the address family. This is always `AF_INET`, or `02`, as defined in the header file `/usr/include/sys/socket.h`.
  - B. The second byte is always `00`.
  - C. The third and fourth bytes are the hexadecimal representation of the port on which *guardian* listens for UUCP requests. In our case, decimal 698 equals `2ba` in hexadecimal, hence the third byte is `02`, and the fourth byte is `ba`.
  - D. The fifth, sixth, seventh, and eighth bytes represent the hexadecimal equivalent of each octet of the IP number of *guardian*. If the IP number of *guardian* is `132.147.144.77`, then the fifth, sixth, seventh, and eighth bytes are `84`, `93`, `90`, and `4d` respectively.
  - E. The last eight bytes are all `00`.
  - F. Putting it all together, the address constructed is:

```
02|00|02|ba|84|93|90|4d|00|00|00|00|00|00|00|00
```

for a total of 16 bytes. The bars (|) are just for clarity.

- G. To create the octal representation of the address, convert each hexadecimal byte above into octal:

```
\002\000\002\272\204\223\220\115\000\000\000\000\000\000\000\000
```

3. Set up the Network Listening Service as follows on *guardian*:

- A. As *root*, execute:

```
nlsadmin -i inet/tcp
```

This creates the directory */usr/net/nls/inet/tcp*, and allows us to add services that listen on the device */dev/inet/tcp*.

- B. Execute:

```
nlsadmin -a 101 -c "/usr/lib/uucp/uucico -r0 -iTLI -unuucp"
-w uucp -y "TLI UUCP" inet/tcp
```

This command should be all on one line. (Do not use a backslash.) This adds service number 101, which executes the command */usr/lib/uucp/uucico -r0 -iTLI -unuucp* as the user *uucp*, when a request for service number 101 is received on the device */dev/inet/tcp*.

The options and arguments to *nlsadmin* are:

|                     |                                                                                                                               |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>-a 101</b>       | Add a service with label 101. (Service code)                                                                                  |
| <b>-c "command"</b> | Fork the <i>command</i> in quotes                                                                                             |
| <b>-w uucp</b>      | Execute the command given in the <b>-c</b> option as the user <i>uucp</i>                                                     |
| <b>-y "TLIUUCP"</b> | Comment                                                                                                                       |
| <b>inet/tcp</b>     | This service is to listen on <i>/dev/inet/tcp</i> . This step requires that the command in step 3A has already been executed. |

The options to the *uucico* command in the **-c** option of *nlsadmin* are:

|                 |                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------|
| <b>-r0</b>      | Act in slave mode                                                                                               |
| <b>-iTLI</b>    | Use the TLI interface                                                                                           |
| <b>-u nuucp</b> | Start up as the user <i>nuucp</i> . This is different from the <b>-w</b> option in the <i>nlsadmin</i> command. |

- C. Execute the command:

```
nlsadmin -l "\x020002ba8493904d0000000000000000" inet/tcp
```

This is the hexadecimal representation of the address constructed in step 2.

4. Add the following line to */usr/lib/uucp/Systems* on *guardian*:

```
colossus Never TLI,e Any
```

If this line for *colossus* has already been added for TLI so that *guardian* can initiate a connection, make no changes to the */usr/lib/uucp/Systems* file.

5. Add the following set of lines to */usr/lib/uucp/Permissions* on *guardian*:

```
MACHINE=colossus LOGNAME=nuucp \
COMMANDS=rmail:rnews:uucp \
READ=/usr/spool/uucppublic:/usr/tmp \
WRITE=/usr/spool/uucppublic:/usr/tmp \
SENDFILES=yes REQUEST=yes
```

6. Add the following line to the */usr/lib/uucp/Systems* file on *colossus*:

```
guardian Any TLI,e Any address
```

where *address* is the octal representation of the address constructed in step 2. Thus the line to add is:

```
guardian Any TLI,e Any \002\000\002\272\204\223\220\115
\000\000\000\000\000\000\000\000
```

This should all be on one line. The backslashes (\) surrounding each octal byte are required.

7. Add the following line to the */usr/lib/uucp/Devices* file on *colossus*:

```
TLI inet/tcp - Any TLI \D nls
```

8. Add the following line to the */usr/lib/uucp/Dialers* file on *colossus*:

```
nls * * * NLPS:000:001:101:\N\c
```

Note that the "101" specified here matches the service code added with **nlsadmin -a** in step 3B.

9. Add the following lines to the */usr/lib/uucp/Permissions* file on *colossus*:

```
MACHINE=guardian LOGNAME=nuucp \
COMMANDS=rmail:rnews:uucp \
READ=/usr/spool/uucppublic:/usr/tmp \
WRITE=/usr/spool/uucppublic:/usr/tmp \
SENDFILES=yes REQUEST=yes
```

10. Verify that each machine is listed in the other's */etc/hosts* file or resolvable by the nameserver. Add the following lines to the end of the file */etc/rc2.d/S87USRDAEMON* on *guardian*, so that the listener is started when the system is booted:

```
nlsadmin -s inet/tcp 2>/dev/null
[$? = 0] && echo "Started listener"
```

11. Shutdown and reboot both systems. Verify that the process called "listen" is running on *guardian*. You should now be able to issue uucp requests for *guardian* from *colossus*.

## *The nlsadmin(ADM) command*

The `nlsadmin` command described in this section does the following:

- Creates directory `/usr/net/nls/inet/tcp` (step 3A), and creates files called `dbf`, `addr`, and `lock` in this directory.
- Adds the following line to the `dbf` file in the directory `/usr/net/nls/inet/tcp` (step 3B):

```
101:n:uucp:reserved:NULL,./usr/lib/uucp/uucico -r0 -iTLI -uuucp* #TLI UUCP
```

See the comments in this file for an explanation of the fields in this entry.

- Adds the following line to the file `addr` in the directory `/usr/net/nls/inet/tcp` (step 3C):

```
\x020002ba8493904d0000000000000000
```

Port numbers greater than 255 are split across two bytes. In our example, the port being used was 698. In the hexadecimal representation, this port was `021ba`, in two bytes. We do a byte by byte conversion to octal. `02` in octal is `\002`, and `ba` is `\272`, hence we have `\002\272`, not `\001\272`, which would be a direct conversion from decimal. Port numbers less than 256 should have the first byte in both hexadecimal and octal equal to 0. For example, port 253 is `001fd` in hexadecimal, and `\000\375` in octal.

See `nlsadmin(ADM)` for more information.

## *Connecting two local systems using a direct wire*

This section describes how to install a direct wire between two computers. If you are using UUCP to connect remote machines, you can skip this section. To connect two computers with a direct wire, you need to do the following:

- Choose a serial port on each machine.
- Connect a serial wire (RS-232) between the two machines, using the chosen serial ports.
- Edit the UUCP configuration files.

### *Choosing a serial port*

On each machine, you must choose the RS-232 serial port (`/dev/ttyn`) you want to use. If there are no ports available, you must install a new serial line or make one available by removing any device connected to it. If you remove a terminal, make sure no one is logged in.

Find the name of the device special file associated with the line. The device name should have the form:

**`/dev/tty $nn$`**

where  $nn$  is the number of the corresponding line. For example, `/dev/tty1a` usually corresponds to COM1. You need the name of the actual line for later steps. Be sure and use the non-modem control port (for example, `/dev/tty1a` instead of `/dev/tty1A`).

The serial port should be owned by `uucp`. To make sure the line is owned by `uucp`, enter this command:

**`chown uucp /dev/tty $nn$`**

where  $nn$  is the number of the corresponding line.

### ***Connecting a serial cable***

You connect two computers together using an RS-232 cable. The actual pin configurations sometimes vary between machines.

The cable should connect pins 2, 3, and 7 on one computer to the same pins on the second computer. Typically, the cable must be *nulled*, which means that pin 2 on one machine is connected to pin 3 on the other, and vice versa. Because the connections can vary, check the hardware manuals for each computer to determine the proper pin connections.

### ***Edit the UUCP configuration files***

You should edit the UUCP files as instructed in "Editing the UUCP control files" (page 666), except you should use the keyword **Direct** instead of **ACU** in the *Systems* files.

### ***Testing a connection***

The following steps will enable you to use the `cu` command on the main machine to log in to the remote machine.

1. Enable the port of the remote machine:  
**`enable /dev/tty $nn$`**   
where  $nn$  is the serial port you are using.
2. Disable the port of the main machine:  
**`disable /dev/tty $nn$`**   
where  $nn$  is the serial port you are using.

To connect to the remote machine from the main machine, type:

**`cu -x9 -ltty $nn$  dir`**

where  $nn$  is the port of the main machine.

Press the (Bksp) key until the lines are synchronized and you get a login prompt. To log in, you must have an account on the remote machine.

If you do not get a login prompt, read the `cu -x9` output for any clues of what could be wrong. Consult “UUCP error messages” (page 704) for a list of messages.

## Complete UUCP examples

---

This section includes two complete working examples of a UUCP system and the database files.

### Example 1: system gomer

---

The following system (*gomer*) has:

- 1200 baud modem on *tty2B*
- direct connection to system (*poker* - configured for dial-in) on *tty2d* for call-out only
- three valid UUCP logins:

*nuucp*    The public login for email. No password required.

*ubarn*    The on-site login for system (*poker*).

*upay4*    The private login for email and file transfers.

All lines beginning with “#” are comments and are not required. Most examples are partial listings and may contain other entries. The modem answers at 1200 baud first and is set up for both call in and out.

**NOTE** The lines from */etc/passwd* are included here for information only. You must use the `sysadmsh(ADM)` Accounts ⇄ User ⇄ Create or Accounts ⇄ User ⇄ Modify selections to create or alter UUCP login accounts.

#### */etc/passwd*

```
uucp::5:5:Uucp admin:/usr/lib/uucp:
nuucp::201:5:public:/usr/spool/uucplogins/nuucp:/usr/lib/uucp/uucico
upay4::202:5:private:/usr/spool/uucplogins/upay4:/usr/lib/uucp/uucico
ubarn::*:203:5:poker:/usr/spool/uucplogins/ubarn:/usr/lib/uucp/uucico
```

#### */etc/group*

```
uucp:x:5:uucp,nuucp,ubarn,upay4
```

## *letclsystemid*

gomer

## *letclinittab*

```
Se2B:23:respawn:/etc/getty -t60 tty2B 2
Se2b:23:off:/etc/getty tty2b m
Se2D:23:off:/etc/getty -t60 tty2D 2
Se2d:23:off:/etc/getty tty2d m
```

## *lsr/lib/uucp/Devices*

```
300-1200 baud hayes 1200 baud modem.
The Direct tty4b entry is for programming the modem.
ACU tty2B - 300-1200 dialHA12
Direct tty2b - 300-1200 direct
poker tty2d - 9600 direct
```

## *lsr/lib/uucp/Permissions*

```
Public uucp login for mail only.
Can send mail, transfer files to/from uucppublic, and get
a directory (ls) listing.
LOGNAME=nuucp MACHINE=OTHER \
 COMMANDS=rmail:ls:uucp \
 READ=/usr/spool/uucppublic:/usr/tmp \
 WRITE=/usr/spool/uucppublic:/usr/tmp \
 SENDFILES=yes REQUEST=yes
Private uucp login for mail and file transfer.
Only dingbat, ogre, grinch, ... can use this login.
LOGNAME=upay4 VALIDATE=dingbat:ogre:grinch:blitzen \
 COMMANDS=rmail:ls:uucp:who:uux \
 READ=/ WRITE=/ \
 NOREAD=/etc \
 SENDFILES=yes REQUEST=yes
Local trusted connection to gomer
Only poker can use this login.
LOGNAME=ubarn VALIDATE=poker \
 COMMANDS=ALL \
 READ=/ WRITE=/ \
 SENDFILES=yes REQUEST=yes
```

## */usr/lib/uucp/Systems*

```
local calls
dingbat Any ACU 1200 4444444 ogin:-BREAK-ogin:-BREAK-ogin:
 uubig word: wetrot
long distance (evening calls only)
grinch Any1800-0700 ACU 1200 18888888 "" \r ogin:-BREAK-ogin:
 -BREAK-ogin:nuucp
uunet Any1800-0700 ACU 1200 17031111111 ogin:-BREAK-ogin:
 -BREAK-ogin:xytpq sword: grm5q
systems that call in as nuucp (for mail) but NOT call out.
daboss Never
sales Never
guru2 Nevea
poker Never
ogre Never
blitzen Never
```

### *Example 2: system dingbat*

---

The following system (*dingbat*) has:

- 2400 baud modem on *tty1A*
- two valid UUCP logins:

*nuucp*      The public login for email. No password required.

*uubig*      The private login for email and file transfers.

All lines beginning with “#” are comments and are not required. Most examples are partial listings and may contain other entries. The modem answers at 2400 baud first and is set up for both call in and out.

## */etc/passwd*

```
uucp*:5:5:Uucp admin:/usr/lib/uucp:
nuucp*:201:5:public:/usr/spool/uucplogins/nuucp:/usr/lib/uucp/uucico
uubig*:202:5:private:/usr/spool/uucplogins/uubig:/usr/lib/uucp/uucico
```

## */etc/group*

```
uucp:x:5:uucp,nuucp,uubig
```

## */etc/systemid*

```
dingbat
```



#### *letc/inittab*

```
SelA:2:respawn:/etc/getty -t60 tty1A 3
SelA:2:off:/etc/getty tty1A m
```

#### *usr/lib/uucp/Devices*

```
300-2400 baud hayes 2400 baud modem.
The Direct entry is for programming the modem.
ACU tty1A - 300-2400 dialHA24
Direct tty1A - 300-2400 direct
```

#### *usr/lib/uucp/Permissions*

```
Public uucp login for mail only.
Can send mail, transfer files to/from uucppublic, and get
a directory (ls) listing.
LOGNAME=nuucp MACHINE=OTHER \
 COMMANDS=rmail:ls:uucp \
 READ=/usr/spool/uucppublic:/usr/tmp \
 WRITE=/usr/spool/uucppublic:/usr/tmp \
 SENDFILES=yes REQUEST=yes
Private uucp login for mail and file transfer.
Only ogre, grinch, ... can use this login.
LOGNAME=uubig VALIDATE=ogre:grinch:gomer:blitzen \
 COMMANDS=rmail:ls:uucp:who:uux \
 READ=/ WRITE=/ \
 NOREAD=/etc \
 SENDFILES=yes REQUEST=yes
```

#### *usr/lib/uucp/Systems*

```
local calls
gomer Any ACU 2400 3333333 ogin:-BREAK-ogin:-BREAK-ogin:
 upay4 word: dryrot
long distance (evening calls only)
grinch Any1800-0700 ACU 2400 18888888 ** \r ogin:
 -BREAK-ogin:-BREAK-ogin: nuucp
systems that call in as nuucp (for mail) but NOT call out.
daboss Never
damgr Never
guru2 Never
ogre Never
blitzen Never
```

## Sample commands

---

Send mail to another system and have it send the mail back.

**mail othersystem!mysystem!mylogin** (Bourne/Korn shell)

**mail othersystem\!mysystem\!mylogin** (C shell)

**NOTE** MMDF must be configured before you can send mail via UUCP.

Send the file */usr/spool/uucppublic/data* to another system.

**uucp ~/data othersystem!~/** (Bourne/Korn shell)

**uucp \~/data othersystem\!\~/** (C shell)

Display the systems you can call:

**uuname**

Force a call to another system and save the debug output in background:

**/usr/lib/uucp/uttry -r -x9 system**

*system* is the name of the system you want to debug. The output is saved to a file with the same name as the system in *tmp* directory.

## Administering your UUCP system

---

This section discusses the various shell scripts that supervise and maintain UUCP. Consult "Administrative programs" (page 661) for commands available to the system administrator. Included is an extended description of the */usr/spool/uucp* work directory and a special subsection on troubleshooting.

### UUCP maintenance shell scripts

---

There are several aspects of system operation that are governed by shell scripts running as daemons:

- checking the UUCP directory for work (**uudemon.hour**)
- polling of sites that are passive or do not originate calls (**uudemon.poll2**)
- sending of status information to the UUCP administrator (**uudemon.admin**)
- cleaning of the UUCP spool directory (**uudemon.clean**)

These scripts must be set up by the system administrator. See **uudemon(ADM)** for complete instructions.

## Generating log reports on usage: *uulog*

---

The **uulog** program displays log information on UUCP usage according to remote machine. All usage of the programs **uucp**, **uuto**, and **uux** are logged in special log files, one per machine.

### *uulog options*

The **uulog** command has the following options:

- fsystem** displays the last entry or entries of the *system* file transfer log.
- ssystem** displays the *system* file transfer information.
- xsystem** displays the **uuxqt** log file for the given system.
- number** specifies the *number* of lines displayed by the **-f** option.

For example, to print the last 10 lines of *chicago*'s file-transfer log, you would enter:

```
uulog -fchicago -10
```

### *Special uulog files*

During execution of the **uulog** program, the files from the following directories are examined:

- /usr/spool/uucp/.Log/uucico/\**  
directory used for log files by the **uucico** program
- /usr/spool/uucp/.Log/uuxqt/\**  
directory used for log files by the **uuxqt** program

## *The UUCP spool directory*

---

The following is a comprehensive discussion of all files and subdirectories of the UUCP spool directory. These files are created in spool directories to lock devices, hold temporary data, or keep information about remote transfers or executions.

### TM. (temporary data file)

These data files are created by UUCP processes under the spool directory (that is, */usr/spool/uucp/system*) when a file is received from another computer. The *system* directory has the same name as the remote computer that is sending the file. The names of the temporary data files have the format:

*TM.pid.ddd*

where *pid* is a process-ID and *ddd* is a sequential three-digit number starting at 0.

When the entire file is received, the *TM.pid.ddd* file is moved to the path-name specified in the *C.sysnxxxx* file (discussed below) that caused the transmission. If processing is abnormally terminated, the *TM.pid.ddd* file may remain in the *system* directory. These files should be automatically removed by *uuclean*.

#### LCK. (lock file)

Lock files are created in the */usr/spool/uucp* directory for each device in use. Lock files prevent duplicate conversations and multiple attempts to use the same calling device. The names of lock files have the format:

*LCK..str*

where *str* is either a device or computer name. These files may remain in the spool directory if the communications link is unexpectedly dropped (usually on computer crashes). The lock files are ignored (removed) after the parent process is no longer active. The lock file contains the process ID of the process that created the lock. The lock file is always named by converting the last letter to lowercase (meaning non-modem control) to avoid possible conflicts if the same line is specified both as modem-control and non-modem-control. For example, the lock on */dev/tty1A* is named *LCK..tty1a*.

#### C. (work file)

Work files are created in a spool directory on the local computer when work (file transfer or remote command execution) is queued for a remote computer. The names of work files have the format:

*C.sysnxxxx*

where *sys* is the name of the remote computer, *n* is the ASCII character representing the grade (priority) of the work, and *xxxx* is the four-digit job sequence number assigned by UUCP. Work files contain the following information:

- full pathname of the file to be sent or requested
- full pathname of the destination or *~/filename*; "*~*" is shorthand for */usr/spool/uucppublic* and must be included if the full pathname is not used
- user login name
- list of options
- name of associated data file in the spool directory. If the *uucp -c* or *uuto -p* option was specified, a dummy name (*D.0*) is used
- mode bits of the source file
- remote user's login name to be notified upon completion of the transfer

### D.(data file)

Data files are created in the spool directory on both the local and remote computers when it is specified in the command line to copy the source file to the spool directory. The names of data files have the following format:

*D.systemxxxxyyy*

where *system* is the first five characters in the name of the remote computer, *xxxx* is a four-digit job sequence number assigned by *uucp*. The four-digit job sequence number may be followed by a subsequence number, *yyy*, that is used when there are several *D.* files created for a work (C.) file.

### X. (execute file)

Execute files are created in the spool directory on the remote computer prior to remote command executions. The names of execute files have the following format:

*X.sysnxxxx*

where *sys* is the name of the remote computer, *n* is the character representing the grade (priority) of the work, and *xxxx* is a four-digit sequence number assigned by UUCP. Execute files contain the following information:

- requester's login and computer name
- name of file(s) required for execution
- input to be used as the standard input to the command string
- computer and file name to receive standard output from the command execution
- command string
- option lines for return status requests

## *UUCP error messages*

---

This section lists the error messages associated with UUCP. There are two types of error messages. **ASSERT** errors are recorded in the */usr/spool/uucp/.Admin/errors* file. **STATUS** errors are recorded in individual machine files found in the */usr/spool/uucp/.Status* directory. Error messages are also stored in */usr/spool/uucp/.Log/uucico/sysname*.

## ASSERT error messages

When a process is aborted, ASSERT error messages are recorded in */usr/spool/uucp/.Admin/errors*. These messages include the filename, SCCS ID, line number, and the text listed in these messages. In most cases, these errors are the result of filesystem problems. Use **errno** (when present) to investigate the problem. If **errno** is present in a message, it is shown as "()" in this list.

**Table 13-4** ASSERT error messages

| Error message               | Description or action                                                                                                                                              |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CAN'T OPEN                  | An <b>open()</b> or <b>fopen()</b> failed. Check for the presence of the file and incorrect permissions.                                                           |
| CAN'T WRITE                 | A <b>write()</b> , <b>fwrite()</b> , <b>fprint()</b> , and so on failed. Check for the presence of the file and incorrect permissions.                             |
| CAN'T READ                  | A <b>read()</b> , <b>fgets()</b> , and so on failed. Check for the presence of the file and incorrect permissions.                                                 |
| CAN'T CREATE                | A <b>create()</b> call failed. Check permissions.                                                                                                                  |
| CAN'T ALLOCATE              | A dynamic allocation failed.                                                                                                                                       |
| CAN'T LOCK                  | An attempt to make a LCK (lock) file failed. In some cases, this is a fatal error.                                                                                 |
| CAN'T STAT                  | A <b>stat()</b> call failed. Check for the presence of the file and incorrect permissions.                                                                         |
| CAN'T CHMOD                 | A <b>chmod()</b> call failed. Check for the presence of the file and incorrect permissions.                                                                        |
| CAN'T LINK                  | A <b>link()</b> call failed. Check for the presence of the file and incorrect permissions.                                                                         |
| CAN'T CHDIR                 | A <b>chdir()</b> call failed. Check for the presence of the file and incorrect permissions.                                                                        |
| CAN'T UNLINK                | An <b>unlink()</b> call failed.                                                                                                                                    |
| WRONG ROLE                  | This is an internal logic problem.                                                                                                                                 |
| CAN'T MOVE TO<br>CORRUPTDIR | An attempt to move some bad C. or X. files to the <i>/usr/spool/uucp/.Corrupt</i> directory failed. The directory is probably missing or has wrong modes or owner. |

(Continued on next page)

**Table 13-4 ASSERT error messages**  
(Continued)

| Error message                   | Description or action                                                                                                                                                     |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CAN'T CLOSE                     | A <code>close()</code> or <code>fclose()</code> call failed.                                                                                                              |
| FILE EXISTS                     | The creation of a C. or D. file is attempted, but the file exists. This occurs when there is a problem with the sequence file access. Usually indicates a software error. |
| No uucp server                  | A TCP/IP call is attempted, but there is no server for UUCP.                                                                                                              |
| BAD UID                         | The uid cannot be found in the <i>/etc/passwd</i> file. The filesystem is in trouble, or the <i>/etc/passwd</i> file is inconsistent.                                     |
| ULIMIT TOO SMALL                | The <code>ulimit</code> for the current user process is too small. File transfers may fail, so transfer is not attempted.                                                 |
| BAD LINE                        | There is a bad line in the <i>Devices</i> file; there are not enough arguments on one or more lines.                                                                      |
| FSTAT FAILED<br>IN EWRDATA      | There is something wrong with the Ethernet media.                                                                                                                         |
| SYSLST OVERFLOW                 | An internal table in <i>gename.c</i> overflowed. A big or strange request was attempted.                                                                                  |
| TOO MANY SAVED<br>C FILES       | Same as previous message.                                                                                                                                                 |
| RETURN FROM<br>fixline ioctl    | An <code>ioctl</code> , which should never fail, failed. There is a system driver problem.                                                                                |
| BAD SPEED                       | A bad line speed appears in the <i>Devices</i> or <i>Systems</i> file ("Class" field).                                                                                    |
| PERMISSIONS file:<br>BAD OPTION | There is a bad line or option in the <i>Permissions</i> file.                                                                                                             |
| PKCGET READ                     | The remote machine probably hung up. No action need be taken.                                                                                                             |
| PKXSTART                        | The remote machine aborted in a non-recoverable way. This can generally be ignored.                                                                                       |

(Continued on next page)

**Table 13-4 ASSERT error messages**  
(Continued)

| Error message    | Description or action                                                                                                                                                                    |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SYSTAT OPEN FAIL | There is a problem with the modes of <i>/usr/lib/uucp/.Status</i> , or there is a file with bad modes in the directory.                                                                  |
| TOO MANY LOCKS   | There is an internal problem!                                                                                                                                                            |
| XMV ERROR        | There is a problem with some file or directory. It is probably the spool directory, because the modes of the destinations were supposed to be checked before this process was attempted. |
| CAN'T FORK       | An attempt to fork and exec failed. The current job should not be lost, but is attempted later ( <b>uuxqt</b> ). No action need be taken.                                                |

## ***UUCP STATUS error messages***

UUCP STATUS error messages are messages that are stored in the */usr/spool/uucp/.Status* directory. This directory contains a separate file for each remote machine that your system attempts to communicate with. These individual machine files contain status information on the attempted communication, and whether it was successful or not. What follows is a list of the most common error messages that can appear in these files.



**Table 13-5 STATUS error messages**

| Error message                 | Description or action                                                                                                                                                                                                            |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OK                            | Self explanatory.                                                                                                                                                                                                                |
| NO DEVICES AVAILABLE          | There is currently no device available for the call. Check to see that there is a valid device in the <i>Devices</i> file for the particular system. Check the <i>Systems</i> file for the device to be used to call the system. |
| WRONG TIME TO CALL            | A call was placed to the system at a time other than what is specified in the <i>Systems</i> file.                                                                                                                               |
| TALKING                       | Self explanatory.                                                                                                                                                                                                                |
| LOGIN FAILED                  | The login for the given machine failed. It could be a wrong login or password, wrong number, a very slow machine, or failure in getting through the <i>dialer-token</i> script.                                                  |
| CONVERSATION FAILED           | The conversation failed after successful startup. This usually means that one side went down, the program aborted, or the line (link) was dropped.                                                                               |
| DIAL FAILED                   | The remote machine never answered. It could be a bad dialer or the wrong phone number.                                                                                                                                           |
| BAD LOGIN/MACHINE COMBINATION | The machine called us with a login or machine name that does not agree with the <i>Permissions</i> file. This could be an attempt to masquerade!                                                                                 |
| DEVICE LOCKED                 | The calling device to be used is currently locked and in use by another process.                                                                                                                                                 |
| ASSERT ERROR                  | An <b>ASSERT</b> error occurred. Check the <i>/usr/spool/uucp/.Admin/errors</i> file for the error message and refer to "UUCP error messages" (page 704).                                                                        |
| SYSTEM NOT IN <i>Systems</i>  | The system is not in the <i>Systems</i> file.                                                                                                                                                                                    |
| CAN'T ACCESS DEVICE           | The device tried does not exist or the modes are wrong. Check the appropriate entries in the <i>Systems</i> and <i>Devices</i> files.                                                                                            |

(Continued on next page)

**Table 13-5 STATUS error messages**  
(Continued)

| Error message                  | Description or action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DEVICE FAILED                  | The open of the device failed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| WRONG MACHINE NAME             | The called machine is reporting a different name than expected.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| CALLBACK REQUIRED              | The called machine requires that it calls your system.                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| REMOTE HAS A LCK FILE FOR ME   | The remote site has a LCK file for your system. They could be trying to call your machine. If they have an older version of UUCP, the process that was talking to your machine may have failed leaving the LCK file. If they have the new version of UUCP and they are not communicating with your system, then the process that has a LCK file is hung. This can also be caused by incorrect permissions in the <i>/usr/spool/uucp</i> path on the remote system, or cleared <i>uucico</i> SUID bit. |
| REMOTE DOES NOT KNOW ME        | The remote machine does not have the node name of your system in its <i>Systems</i> file.                                                                                                                                                                                                                                                                                                                                                                                                             |
| REMOTE REJECT AFTER LOGIN      | The login used by your system to log in does not agree with what the remote machine was expecting. Check the <i>Permissions</i> file on the remote system.                                                                                                                                                                                                                                                                                                                                            |
| REMOTE REJECT, UNKNOWN MESSAGE | The remote machine rejected the communication with your system for an unknown reason. The remote machine may not be running a standard version of UUCP.                                                                                                                                                                                                                                                                                                                                               |
| STARTUP FAILED                 | Login succeeded, but initial handshake failed. Check communication parameters: data word size, parity, stop bits, and so on.                                                                                                                                                                                                                                                                                                                                                                          |
| CALLER SCRIPT FAILED           | This is usually the same as DIAL FAILED. However, if it occurs often, suspect the caller script in the <i>Dialers</i> file. Use <i>uutry</i> to check.                                                                                                                                                                                                                                                                                                                                                |

## Chapter 14

# Setting up electronic mail

---

This chapter explains how to set up electronic mail on your system.

Electronic mail on SCO Open Desktop/SCO Open Server is handled by two utilities, the MAIL USER AGENT (MUA) and the MAIL TRANSPORT AGENT (MTA). The MUA is the program, such as `mail(C)`, that allows users to send, read, and manage mail messages. The MUA transfers the message to the MTA, the group of programs that actually route and deliver messages to their destinations. The MTA on the SCO UNIX operating system is MMDF (Multichannel Memorandum Distribution Facility).

MMDF provides users with transparent access to the different networks and related mail transport PROTOCOLS, through CHANNELS, regardless of the MUA. (A channel is the method, such as UUCP, used to deliver messages; a protocol is a set of rules for communicating over a network and includes standards for mail message formats.)

In addition, MMDF provides the system administrator with tools to monitor and customize MMDF. Using these tools, the system administrator can tune MMDF dynamically, modifying the behavior of these programs even while MMDF is running.

With MMDF, users can send mail on the local network or over larger area networks across the interconnected group of networks known as the "Internet". The Internet is otherwise known as ArpaNet or DARPA (Defense Advanced Research Projects Agency) Internet.

The version of SCO MMDF provided with your operating system was derived from MMDF-II Release 43 from the University of Delaware. This version provides additional features, such as a more robust locking mechanism that allows you to specify how to lock user mailboxes, and support for `sendmail.forward` files. This version also differs in the format of the hashed table database built by `dbmbuild`(ADM), which is nonstandard.

The system automatically configures MMDF for local (one system) mail delivery when you install your operating system. If you did not install the entire distribution, you should install the MAIL package now using the `custom`(ADM) utility. See your *Installation and Upgrade Guide* information on installing packages with `custom`.

## Chapter overview

---

The first part of this chapter explains how to configure MMDF for most sites using the simple configuration utility provided with your distribution. First, the chapter gives a basic overview of how MMDF works and covers some background information necessary for using the configuration utility. With this information, you can then fill out the MMDF configuration checklist provided. Then, this chapter shows you how to run the configuration utility to set up MMDF to exchange mail with other computers.

The configuration utility does not work for all site configurations, therefore the second part of this chapter describes the different configuration files, how MMDF uses them, and how to modify them to work with your configuration. This section also covers how to test and maintain the MMDF system.

If you do not plan to exchange mail with other machines via UUCP or TCP/IP, you do not have to run the configuration utility; MMDF is already configured to send mail on the local machine. However, if you want to reroute mail sent to special system accounts, such as `root`, the configuration utility provides an easy way for you to set this up. For more information, see "Redirect mail for root?" (page 724).

## How MMDF works

---

This section gives you an overview of how MMDF processes and delivers mail on your UNIX system. Most of the concepts relate to mail traveling in both directions; you should read both the outgoing and incoming sections with this assumption.

## Outgoing mail

---

Outgoing mail starts when the user invokes the MUA, such as **mail(C)**, to compose a mail message. The MUA requires that the user specify a "To:" header when creating the message. Then, the MUA adds two other headers, "Date:" and "From:", when the user sends the message. These headers specify how MMDF sends the message through the system; the next section explains the format of these headers.

### Mail headers

For a mail header to be correct, it must include these three lines in the following format:

```
From: sender
To: recipient
Date: Weekday Mon DD hh:mm:ss year
```

Here is an example:

```
From: fred@npr.COM (Fred Astaire)
To: ginger
Date: Wed Apr 3 12:21:23 PST 1991
```

In addition, most MUAs and mail submission programs, such as **mail(C)** and **execmail**, add extra header lines. For example, the MUA might add the "Message-Id" header. If the MUA does not add this header, MMDF adds it, as well as any "Received" headers. The user can also add other headers. For example, if the user specifies a carbon-copy recipient, the message header includes a "Cc:" line. MMDF allows these additional header lines, but does not use them; the recipient mail server and MUA handle these header lines.

In the example above, note the format of the address in the "From:" line. This is an example of a DOMAIN NAME. MMDF uses the domain name to determine how to route the message. Before you can configure MMDF with the configuration utility, you must understand how domain names work; "MMDF configuration checklist" (page 719) covers domains and the different types of domain names, including fully qualified domain names, in greater detail.

Figure 14-1 shows the path that the message takes through MMDF once you send your message with the MUA.

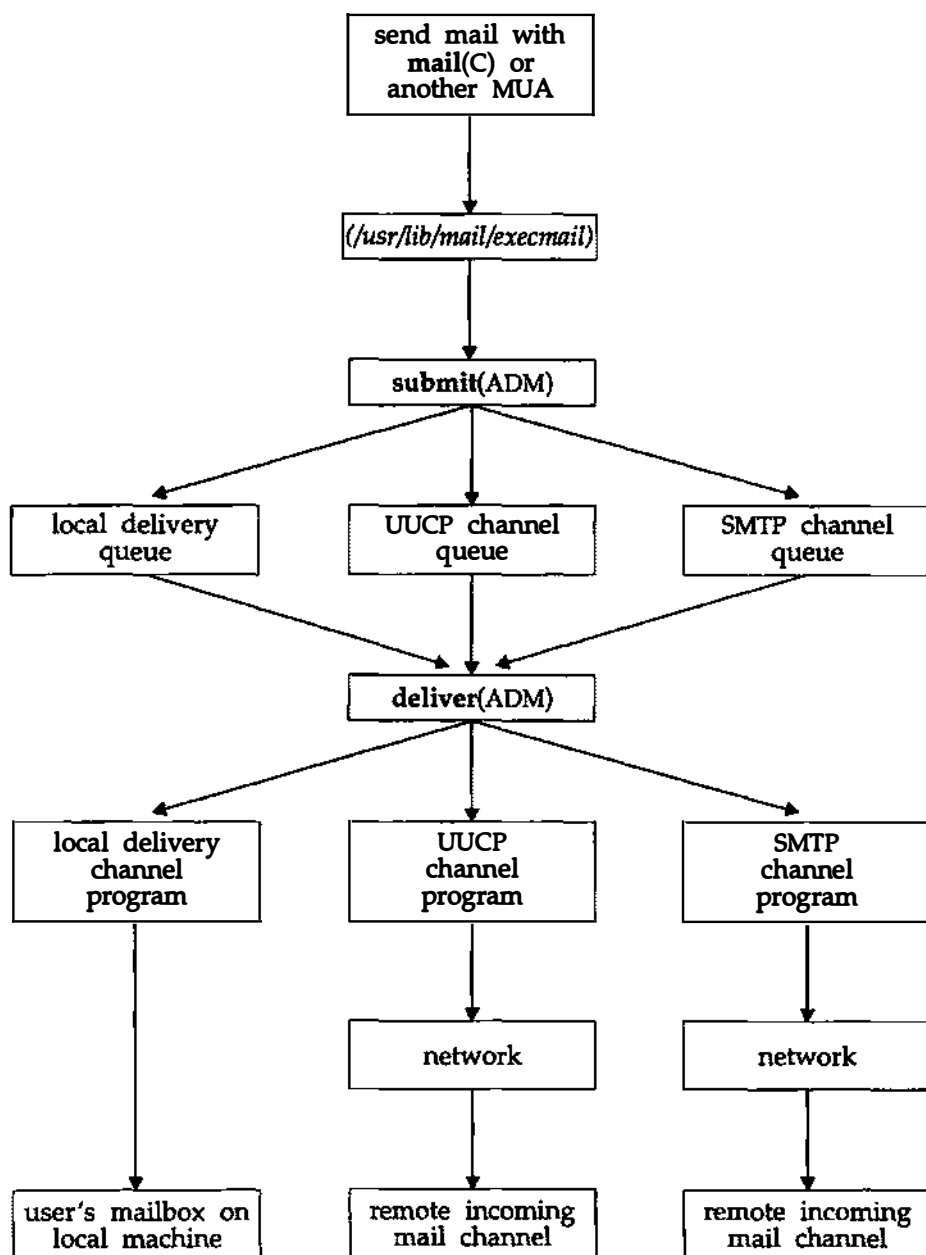


Figure 14-1 Outgoing mail route

The MUA allows the user to create, edit, and start a message on its journey. Then, the MUA transfers the message to `execmail`, which sends it to `submit(ADM)`. The `submit` program uses the information in the "To:" line and the MMDF configuration files to build the fully qualified domain name. From this, `submit` determines the channel to use (for example, UUCP) when sending the message and then places the message in the appropriate channel queue for processing later by the `deliver(ADM)` program. The next time `deliver` runs (by default, `deliver` runs every 10 minutes), it transfers the messages from the channel queue to the appropriate channel program.

For example, you send a message with the following "To:" line:

```
To: andrei@npr.com
```

The `submit` program looks in the appropriate MMDF configuration table for `npr` and builds the fully qualified domain name; for example:

```
scribe.npr.com
```

Now, `submit` determines the appropriate channel to use to send the message. In this case, for example, all messages in the `npr.com` domain are sent to the outside world on the UUCP channel, so `submit` puts the message in the UUCP channel queue. Then `deliver` picks up the message and passes it to the channel program. The channel program transfers the message out of the MMDF mail system and into the UUCP subsystem where it is queued and sent via UUCP.

## Incoming mail

MMDF processes incoming mail in much the same way it processes outgoing mail; Figure 14-2 illustrates how MMDF handles incoming mail.

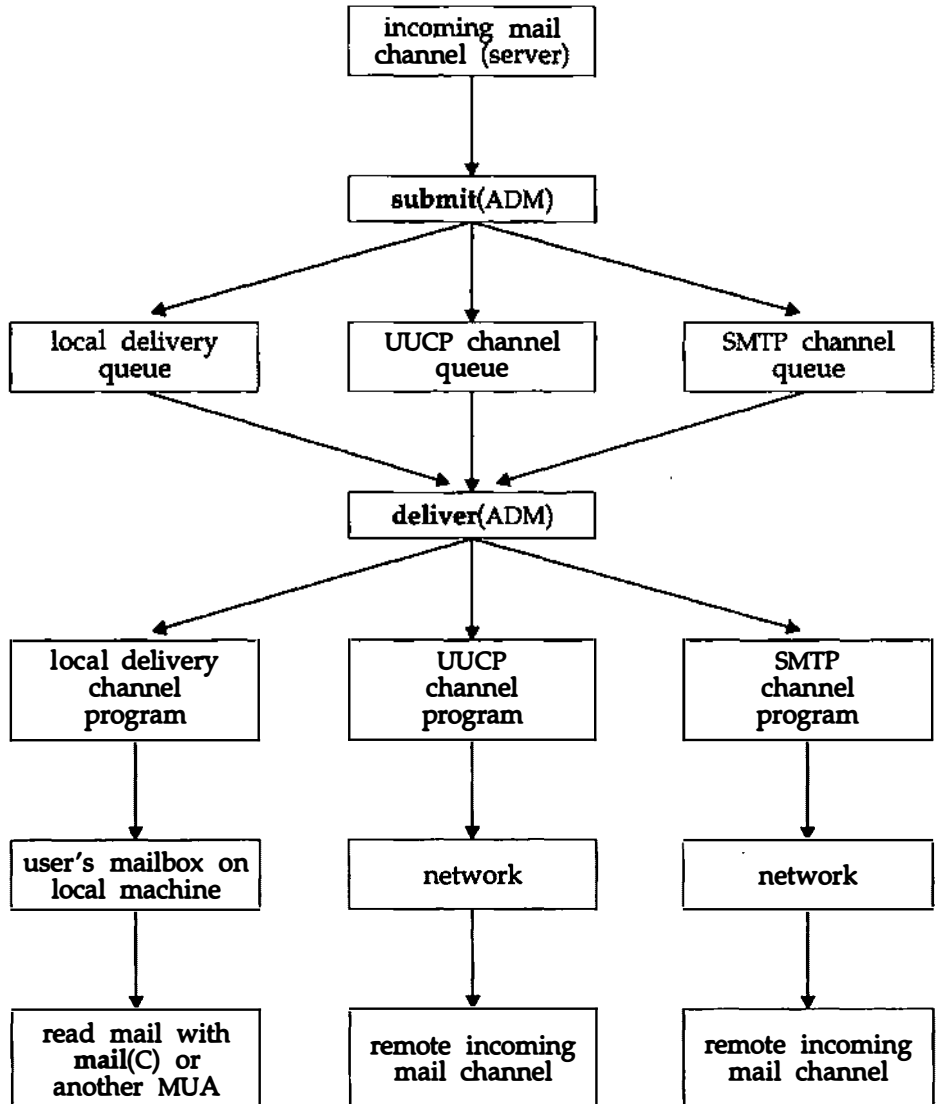


Figure 14-2 Incoming mail



MMDF processes incoming mail exactly as it processes outgoing mail (the same as when the MUA calls `submit(ADM)`) except that MMDF adds a "Received" line.

MMDF receives incoming mail from a channel's input program (called a "server"), such as `rmail(C)` for the UUCP program. The `submit` program determines the destination of the message and then designates the channel to use to deliver the mail. At this point, MMDF generally selects channels such as `local`, `badusers`, and `badhosts`. Mail destined for the local machine goes to the `local` channel; when the machine is local, but the user is not, mail goes to `badusers`; and if it requires further processing, the message goes to the `badhosts` channel. If the message is destined for another machine, such as when the local machine is acting as a gateway between networks, MMDF selects another network channel, such as UUCP.

The `submit` program puts the message in the appropriate queue and the `deliver` program transfers the mail from each queue to the appropriate channel; eventually the message is delivered to the user.

## The configuration utility

---

The configuration utility facilitates MMDF configuration by building the appropriate configuration files at installation time. All sites should use the configuration utility to perform the initial MMDF configuration.

The configuration files that this utility builds are sufficient for most sites. If your site is one that the configuration utility does not handle, after building the initial configuration, you must edit the MMDF configuration files manually using the information in "MMDF configuration files" (page 739) and "Modifying MMDF configuration" (page 759). (See "Configuration utility limitations" (page 718) for a description of the limitations of this utility.)

In addition, if your configuration changes (for example, you add another machine to your network), you must rebuild the configuration files either by rerunning the configuration utility or by editing these files manually.

## *Configuring communications channels*

---

Before you set up MMDF, you must configure all the communications channels that you plan to use to route mail. MMDF supports the following channels:

- UUCP (UNIX-to-UNIX Copy subsystem)
- SMTP (TCP/IP)

The configuration utility configures for all the channels provided by SCO: UUCP, and SMTP. MMDF also supports other third-party mail products that provide channel programs; however, you cannot use the configuration utility to set up MMDF to use these products. See “Advanced MMDF configuration tasks” (page 771) for information on configuring MMDF for use with other channels. For example, if you plan to exchange mail with other hosts via UUCP, you must configure UUCP on the local host before running the configuration utility. For information on configuring UUCP, see the “Building a remote network with UUCP” chapter in this guide.

## *Configuration utility limitations*

---

This section discusses the situations in which the configuration utility builds either an incorrect or incomplete MMDF configuration for your site. If the situation at your site matches one or more of the situations in this list, you must manually update the MMDF configuration files after building the initial configuration with the configuration utility.

- You use more than one communications channel to exchange mail with a particular host.

If your host communicates with another host using more than one channel, the configuration utility sets up MMDF to communicate with that host using the first configured channel in the following order:

1. TCP/IP
2. UUCP

If your host accesses another host using two or more channels, but you do not want to transfer mail using the first applicable channel, the configuration utility does not configure the MMDF correctly. See “Modifying MMDF configuration” (page 759) for more information.

- Your host does not exchange mail with every host on each configured channel.

The configuration utility automatically sets up MMDF to exchange mail with every host on the configured channels. For information on removing hosts from a configured channel, see the sections on modifying the information for that channel in “Modifying MMDF configuration” (page 759).

- If you exchange mail with two hosts and you want to prevent one host from passing mail to the other host through the local host.

The configuration script sets up MMDF to allow one host to send mail through the local host to the other host. If you want to prevent this, you must set up authorization to prevent the transfer. For information on how to do this, see “Configuring MMDF authorization” (page 772).

- You have more than one channel of the same type.

The configuration utility lists all the hosts that your host accesses using a particular communications protocol in the same channel. To set up authorization to restrict a host’s access on a channel, see “Configuring MMDF authorization” (page 772).

If your configuration changes for some reason after running the utility (for example, you might add a new UUCP host), you should reexamine this list. If the utility is still able to set up your site correctly, you can update your configuration by running the utility again; otherwise, you must update the configuration files manually.

## *MMDF configuration checklist*

---

Once you configure the communications channels, use the checklist in this section to write down the information that you need to respond to prompts during the configuration procedure. Items in the checklist that are surrounded by boxes pertain to specific configurations (for example, if you are routing mail over UUCP); fill out these sections only if you have these configurations.

The sections following the checklist contain information on each of the items in the checklist.

### Configuration Checklist

1. Host name: \_\_\_\_\_
2. Domain name: \_\_\_\_\_
3. Fully qualified host name: \_\_\_\_\_
4. Hide your host name? ☐ Yes  
☐ No
5. Redirect mail for *root*?: \_\_\_\_\_
6. Redirect mail for *mmdf*?: \_\_\_\_\_
7. *postmaster* address: \_\_\_\_\_

**If you have UUCP installed and configured and you plan to use MMDF to route mail over UUCP:**

- UUCP host name: \_\_\_\_\_
- UUCP hosts with full domain names? ☐ No  
☐ Yes; enter domain names: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**If you have TCP/IP installed and configured and you plan to route mail over TCP/IP (SMTP):**

- Configure a name server? ☐ Yes  
☐ No

8. Smart host (badhosts): \_\_\_\_\_  
(where to route mail to unrecognized machines)
9. Smart host (badusers): \_\_\_\_\_  
(where to route mail to unrecognized users)

Use the information in the following sections to complete the checklist.

## Host name

---

The HOST NAME (or machine name) is the name of the machine on which you are configuring MMDF. To determine the host name, enter this command at your UNIX system prompt:

```
cat /etc/systemid
```

An example of a host name is *scribe*.

**NOTE** SCO MMDF, and MMDF in general, allow host names to contain underscore characters; this is an extension of internet standards. To ensure interoperability with hosts that strictly conform to internet standards, we recommend that you do not use the underscore character when internet-working is a primary goal.

## Domain name

---

A DOMAIN NAME is the section of a mail address that appears to the right of the at (@) character, for example, *npr.com*. The domain describes the site where your machine is located and generally includes the machine (host) name, a department (optionally), and the site's organization or country. MMDF uses the domain to deliver the message to the appropriate location. Note that the domain name uniquely identifies a machine, but not the path by which messages reach that machine.

The following is the convention for specifying domains:

*hostname.subdomain.top-level*

If the domain includes a department, the convention is:

*hostname.local.subdomain.top-level*

Here is a description of each of the domain levels:

### Top-Level Domain

A top-level domain is an officially registered name that describes the purpose of a group of institutions or a code that is associated with a country.

You can only use registered top-level and subdomain names if you have registered your organization with SRI International. For information on registering your domain, see "Registering domain names" (page 723). If you have not registered with SRI, use the UUCP top-level domain.

In the United States, the common top-level domains on the Internet are:<sup>1</sup>

|             |                                                                                                             |
|-------------|-------------------------------------------------------------------------------------------------------------|
| COM         | commercial institutions                                                                                     |
| <i>code</i> | country code <sup>2</sup>                                                                                   |
| EDU         | educational and research institutions                                                                       |
| GOV         | government institutions                                                                                     |
| MIL         | military institutions                                                                                       |
| NET         | network                                                                                                     |
| ORG         | organization (generic)                                                                                      |
| UUCP        | an unregistered domain name where users transmit information between cooperating neighbor machines via UUCP |

### Subdomain

An officially registered name that describes a company, department, or any subgroup under a top-level domain; *sco* is an example of a subdomain in the domain *COM*.

### Local Domain

A name recognized only within an organization that has meaning only within that organization; a department name such as *enr* is an example of a local domain.

## ***Fully qualified host name***

---

The complete domain name (the machine name and all other domain names) is known as the FULLY QUALIFIED HOST NAME (or fully qualified domain name). For example, if you have this mail address:

`andrei@scribe.npr.com`

the fully qualified domain name is:

`scribe.npr.com`

In this case, *scribe* is the machine name, *npr* is the company, and *com* specifies that the machine belongs to a commercial organization.

---

\* Domain names can be either upper or lowercase; MMDF is case-insensitive when evaluating domain names.

† The International Standards Organization (ISO) standard 3166 defines the country codes. For example, US is the country code for the United States, AU for Australia, DK for Denmark, and JP for Japan.

The following table shows examples of other fully qualified domain names:

|                                     |                                                                                                                                |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <code>slug.ucsc.EDU</code>          | <i>slug</i> is a machine at the University of California at Santa Cruz in the <i>EDU</i> domain                                |
| <code>seismo.css.GOV</code>         | <i>seismo</i> is a machine at the Center for Seismographic Studies in the <i>GOV</i> domain                                    |
| <code>nessie.edinburgh.ac.UK</code> | <i>nessie</i> is a machine at Edinburgh University in Scotland in the academic ( <i>ac</i> ) subdomain in the <i>UK</i> domain |

To the configuration utility, the fully qualified host name is the host name followed by the local (if appropriate), subdomain, and top-level domain name as shown in the previous examples. The domain name refers to just the subdomain and top-level domain name (without the host name). For example, *npr.com*.

## Registering domain names

If you or any user at your site plan to receive or send mail outside of your organization, you should register a top-level domain or subdomain with the InterNIC Registration Services. Even if you are only using UUCP, registration with the InterNIC is your only guarantee that the name of your site is unique.

To register a top-level domain and subdomain name call or write to the NIC standards organization at the following address:

Surface mail:      Network Solutions  
                           Attn: InterNIC Registration Services  
                           505 Huntmar Park Drive  
                           Herndon, VA 22070

Telephone:        1-703-742-4777 (7:00 AM to 7:00 PM EST)

Email address:    `hostmaster@internic.net`  
                           (host, domain, network changes, and updates)  
                           `action@internic.net`  
                           (computer operations)  
                           `mailserv@rs.internic.net`  
                           (automatic mail servers)

Network address: 198.41.0.5 (RS.INTERNIC.NET)

The earlier your site enrolls a domain name and InterNIC Registration Services gives you an address, the less likely it is that you will have to alter a machine name or other site identifier later.

## Hide your host name?

---

If you have several machines at your site (domain) and you register your domain name with the InterNIC you can “hide” your host name behind your domain name. When you configure MMDF to hide your host name, people outside your organization can send mail to people who receive mail on different machines within your organization without having to know the name of those specific machines. In this case, MMDF identifies all outgoing mail as coming from a single source (your registered domain name). In addition, MMDF routes mail sent to this source to its correct destination within your organization.

For example, the fully qualified domain name for the host that you are configuring is:

`scribe.npr.com`

You can “hide” the host name *scribe* behind the domain name *npr.com*. In this case, someone outside the *npr* organization can send mail to *andrei* on *scribe* without specifying the fully qualified domain name. Instead, they can use *andrei@npr.com* as the address.

In addition, when *andrei* sends mail outside the organization, the message appears as if it came from *andrei@npr.com* instead of *andrei@scribe.npr.com*.

If you want to hide the current host behind the registered domain name, press (Return) when the configuration utility prompts you.

**NOTE** When you join local machines under a single domain name, you create an administrative domain. Within an administrative domain, all user names must be unique so that mail can go to any person anywhere within the domain without a local machine name in the mail address.

If you decide to hide the host behind a domain name, you should create ALIASES to map each user in the domain to the machine where they receive mail. For information on how to do this, see “Creating aliases for users” (page 738).

## Redirect mail for root?

---

The system sends mail about any system problems to the *root* user. You can configure MMDF to redirect this mail to another person. For example, if you are the system administrator for a group of machines, you can redirect all mail sent to *root* on those machines to your own system mailbox. With this configuration, you do not need to log into each machine to read *root*’s mail.



If you want to redirect *root*'s mail, press (Return) when prompted by the configuration utility and then enter the address of the person to whom you want MMDF to deliver *root*'s mail. For example, if you are configuring MMDF on *scribe.npr.com* and you want the user *bob@talk.npr.com* to take care of mail to *root*, redirect *root*'s mail to *bob* by entering this at the prompt:

**bob@talk.npr.com**

(You can redirect *root*'s mail to a local user by entering just the user name at the prompt; you do not need to enter the fully qualified domain name for a local user.)

## ***Redirect mail for mmdf?***

---

When problems occur with the mail system, the system sends mail to the *mmdf* user. This account is reserved for administering your mail system; unless you log in as *mmdf* regularly, you might not find out about problems with the system. For this reason, it is a good idea to redirect *mmdf*'s mail to another person. To do this, press (Return) when the configuration utility prompts you and then enter the address of the person to whom you want to deliver *mmdf*'s mail.

## ***Postmaster address***

---

On the Internet, people generally send any inquiries about user and host names to the "postmaster" address in that domain. RFC (Request for Comments) 821/822 requires that every host provide this reserved postmaster mailbox. For these reasons, you should designate a user on the local system or within the domain as the postmaster and define a *postmaster* alias. To set up the alias while running the configuration utility, simply enter the name of the postmaster user.

## ***Setting up MMDF for UUCP***

---

If you are setting up the system to exchange mail with another system using UUCP, the configuration utility prompts you for some information about your UUCP hosts.

### ***UUCP host name***

The configuration utility prompts for the UUCP name for the host that you are currently configuring. The configuration utility gets this information from */etc/systemid* (or *uname*); when you install the operating system, the installation prompts you for the system name. In cases where the UUCP name is different from the host name, you should enter the correct name when prompted.

### *UUCP hosts with full domain names*

The configuration utility asks if any of the UUCP hosts have full domain names. This might be the case if your host is not on the Internet but your site has an agreement with another machine on the Internet to transfer your mail. When people on the Internet send messages to you, they use the *user@machine.domain* address format (instead of the *machine!user* UUCP format). You then use UUCP to connect to that machine and pick up your spooled mail.

For example, your machine *scribe.npr.com* connects to *slug.ucsc.edu* on the Internet. The machine *slug.ucsc.edu* is a UUCP host with a full domain name. In this case, enter "y" when the configuration utility prompts you for UUCP hosts and enter the host name and then the fully qualified domain name at the prompts.

### *Configuring MMDF to use a name server*

---

Versions of MMDF included with your SCO system Release 3.0 and later include NAME SERVER support. A name server is a program running on the network that provides a central database of information, such as Internet addresses and the names of hosts on which people receive mail.

If you want the configuration utility to set up MMDF to use the name server, you must set up the name server before running the utility. See Chapter 4, "Configuring the BIND name server," of this guide for details on setting up the name server. If you already have a name server running, the configuration utility sets up MMDF to use it automatically.

### *Configuring smart hosts*

---

If the machine that you are configuring communicates directly with another machine that has more complete information about the entire mail network, you can set up MMDF to route any mail that it does not recognize to that machine. This machine is known as a "smart host". MMDF recognizes two kinds of smart host: hosts with information about machine names and hosts with information about the users on the network.

#### *Badhosts channel*

If you specify a smart host for the machine names, MMDF routes mail destined for machines that it does not recognize to the **badhosts** channel. All mail directed to this channel is sent to the smart host that you specify via the normal channel (such as UUCP) that you use to communicate with that host.

For example, someone sends mail to *boris@kgb.gov* but the local host does not recognize the machine *kgb.gov*. However, your host uses UUCP to communicate with another machine, *rocky.npr.com*, that has more complete information

about the network. If you configure *rocky.npr.com* as your smart host, MMDF puts the message to *boris@kgb.gov* in the **badhosts** channel and then uses UUCP to deliver it to *rocky.npr.com*. The mail system on *rocky.npr.com* then determines the correct route to send the message to its destination.

### ***Badusers channel***

You can also specify a smart host that contains complete information about all the users on a network. This is useful if you have a large number of people at your site. Instead of maintaining information on all the people at the site on each host, you can maintain this information on one central host (the smart host); each individual host maintains information about the local users only. In this case, any mail addressed to users that the local machine does not recognize is directed to the **badusers** channel. The mail routed on this channel is sent to the smart host via the channel (such as UUCP) that you usually use to communicate with that host.

For example, someone on the local host sends mail to *natasha* but the host does not recognize her as a local user. If you communicate with *moose.npr.com*, a smart host that contains complete user information, MMDF routes the message to this host. This does not mean that *natasha* actually receives her mail on *moose*, just that *moose* has more information about where *natasha* is located.

## ***Running the configuration utility***

---

Once you configure the communications channels (and name server, if appropriate) and complete the configuration checklist, you are ready to run the MMDF configuration utility.<sup>3</sup>

**NOTE** The configuration utility provides common default values for many of the prompts; to accept these values, press (Return) at the prompt.

The configuration procedure varies slightly (from step 11 onwards) depending on whether TCP/IP is installed. The procedure detailed in this section assumes that TCP/IP is not installed. If TCP/IP is installed on your system, follow the instructions down to step 11 and then continue from the section headed "Running the configuration utility with TCP/IP installed" later in this chapter.

Use the following procedure to configure MMDF:

---

\* Remember, if you are only planning to route mail on the local machine, you do not have to run the configuration utility; MMDF is already set up for local mail delivery.

## 14. Setting up electronic mail

1. Log in as *mmdf*.
2. Change to the */usr/mmdf* directory and start the configuration utility by entering the following command

**mkdev mmdf**

You see a screen with information about the limitations of the configuration utility.

3. At the bottom of the screen, you see this prompt:

```
Do you wish to continue the configuration process at this time? [y]
```

If you enter "n" at this prompt, you exit the configuration utility and return to the UNIX system prompt. Press **(Return)** or enter "y" to continue with the configuration.

The configuration utility displays a message about the current version of the software, for example:

```
This machine is running level 43NS MMDP.
```

4. If you have not already set the host name on the machine that you are configuring, you see the following prompt (if you have already set the host name, skip this step):

```
The name of this host has not been configured yet.
What will you be calling this host?
```

Enter the host name at this prompt. You must set the name of the host before configuring MMDP.

5. Now, you see the following prompt where *host.domain* is a fully qualified host name like *scribe.npr.com*:

```
Is your fully qualified host name 'host.domain'? [y]
```

If the fully qualified host name is correct, press **(Return)** and go to the next step. If the host name is incorrect, enter "n". The utility displays this prompt:

```
What is the correct host name? (host)
```

Enter the correct host name (for example, *pubsco*) or press (Return) to select the default host name. You see the prompt for the domain name:

What is the correct domain name? [domain]

There are three possible responses to this prompt: you can press (Return) to accept the default domain name, you can enter a different domain name (for example, *eng*), or you can enter a domain and sub-domain name (for example, *techpubs.eng*).

Now, the utility prompts you to confirm the fully qualified host name:

Is your fully qualified host name 'pubsco.techpubs.eng'? [y]

If the host name is correct, press (Return) and go to the next step; otherwise enter *n* and repeat this step.

If you have specified a fully qualified host name which includes a sub-domain, for example, *pubsco.techpubs.eng*, the prompt shown in step 5 is displayed; if you have specified a fully qualified host name which does not include a sub-domain, for example, *pubsco.eng*, step 5 is bypassed.

6. Now, you see some information about "hiding" the host behind the domain, followed by this prompt:

At many sites, it is common for mail to be addressed as being from "person@site" instead of from "person@machine.site". This allows people to be moved between machines internally without requiring them to notify all their external correspondents about the address change. This configuration does, however, require a complete user alias table (see below) containing mappings from user names to the host that they actually plan to read their mail on. If you are not sure, then you should probably answer "yes" if there are two or more machines in the domain "techpubs.eng", and "no" otherwise.

Do you wish to hide 'pubsco.techpubs.eng' behind 'techpubs.eng'? [y]

To "hide" the host that you are configuring behind another host, press (Return) at this prompt. Otherwise, enter "n".

7. Now, the configuration utility displays some information about creating mail aliases for special accounts, *root*, *mmdf*, and *postmaster*, on your system.

First, the utility prompts you to create an alias for *root*:

Do you wish to have mail for root redirected to a real user? [y]

## 14. Setting up electronic mail

If you want mail addressed to *root* to go to *root*'s system mailbox, enter "n" and go to the next step.

If you want *root*'s mail to go to a different person, press (Return) and you see this prompt:

```
To whom should root's mail be sent?
```

Enter the login name of the person that you want to receive mail directed to *root*.

8. Now, you see this prompt:

```
Do you wish to redirect mail addressed to mmdf? [y]
```

If you do not want to redirect mail to *mmdf*, enter "n" and go to the next step. If you want mail addressed to *mmdf* to go to another person (for example, the system administrator), press (Return).

```
To whom should mmdf's mail be sent? [root]
```

The default for this prompt is the same as the selection made for step 6. For example, if *root* was selected in response to step 6 then *root* is displayed here.

To have *mmdf*'s mail go to *root*'s mailbox, simply press (Return). If you want this mail directed to another person, enter the address of that person.

9. You see this prompt:

```
To whom should mail addressed to postmaster be sent? [mmdf]
```

Again, you can direct any mail sent to *postmaster* to another person, such as *mmdf*. Press (Return) to accept the default or enter a different address.

10. If UUCP is not installed on your system, you see a message like this:

```
UUCP not installed, skipping ...
```

11. Now, you see some information about associating login names for each user to the machines where the users actually read their mail by creating an alias file. See "Creating aliases for users" (page 738) for more information.

The configuration utility also displays some information about converting XENIX aliases to the correct MMDF format. For more information, see “Converting from XENIX” (page 763).

If you have UUCP installed on your system, you see this prompt:

```
Are you going to be using UUCP for mail? [y]
```

If you plan to use UUCP for transferring mail, press (Return).

**NOTE** If you are running the configuration utility with TCP/IP installed on your system, go to “Running the configuration utility with TCP/IP installed” (page 734).

12. Now, you see the prompt for the UUCP host name:

```
Is this host known as 'host' for UUCP? [y]
```

If the host name is correct, press (Return) and go to the next step. Otherwise, enter “n” to display the following prompt:

```
What is this host's UUCP name? [host]
```

Enter the new name and press (Return). You see this prompt again:

```
Is this host known as 'host' for UUCP [y]
```

Press (Return) if the host name is correct, or press “n” to change it.

13. At this point, if the configuration utility detects a UUCP connection to the *sosco* machine at *SCO*, you see the following prompt; if not, go to the next step.

```
uname reports a connection to sosco.
Does this link really exist? [n]
```

## 14. Setting up electronic mail

If your system has a UUCP connection for exchanging mail with *sosco*, enter "y" and go to the next step. If not, enter "n" to display the following message:

```
Deleting sosco from list of mail sites
```

If *sosco* was the only configured UUCP connection, the utility exits and displays the following message:

```
UUCP must be configured before mail
```

In this case, configure your UUCP connections before running `mkdev mmdf` again.

14. Now, the configuration utility prompts you for information about the hosts you communicate with using UUCP:

```
Because UUCP does not maintain information about domain names, it will be
necessary for you to provide the domain names of any of the hosts with
which you communicate via UUCP.
```

```
Do any of your UUCP hosts have full domain names? [n]
```

In most cases, UUCP hosts do not have full domain names, so you can simply press `<Return>` at this prompt.

If any of your UUCP hosts have full domain names, enter "y". You see the prompts for the host names of your UUCP hosts:

```
Enter the UUCP site name (blank to terminate):
```

Enter the host name, for example, *pubsco* (not the fully qualified name) and press `<Return>`.

15. The following screen is displayed:

```
What is pubsco's fully qualified name? [pubsco]
```

Enter the fully qualified name, for example, *pubsco.techpubs.eng* and press `<Return>`.



## 16. You see the following messages:

```

Micnet not configured, skipping ...

TCP/IP not installed, skipping ...

Many sites do not have complete information about the entire mail network,
but rely on another "smarter" host to determine the correct route that
mail messages should follow to reach their destinations. Any mail that
the local machine is incapable of correctly handling is passed to the
smart host for further processing. In MMDP, this is called the "badhost"
channel.

Do you have such a "smart" host? [y]

```

If you enter "n", go to the next step; if you enter "y" the following screen is displayed:

```

What is its name? (q if you have changed your mind)

```

## 17. The following screen is displayed:

```

Another option, which is often used on large sites is to have a central
machine which contains complete knowledge about all the users on the site,
and only maintaining local lists on each machine. MMDP provides the
facility to forward mail containing unrecognised local addresses to a
smarter host which will have a complete user data base (via the "baduser"
channel).

Do you have such a "smart" host? [y]

```

If you enter "n", go to the next step; if you enter "y" the following screen is displayed:

```

What is its name? (q if you have changed your mind) (pubsco.techpubs.eng)

```

18. When you finish, the configuration utility displays information similar to the following as it creates and edits the MMDF configuration files (the messages might be different, depending on your configuration):

```
Creating the mmdftailor file: header, host name info, support
address, alias tables, local domain and channel, SMTP, root

Building the alias tables (mostly empty)
A Mailing list alias table (alias.list) already exists, skipping
A general user alias table (alias.user) already exists, skipping

Building channel files
local, list, SMTP, UUCP, badusers, badhosts

Building the domain tables
local, domain, root
done
building the database
```

The configuration is now complete.

## ***Running the configuration utility with TCP/IP installed***

---

This section assumes that TCP/IP is installed on your system. It continues on from step 10 in the previous section.

11. You see the following messages:

```
Micnet not configured, skipping ...

Are you going to be using SMTP for mail? [y]
```

If you enter "n" go to step 13; if you enter "y" go to step 12.

12. The configuration utility displays one of the following messages:

- If the name server is already configured on your system, you see the following screen:

```
A domain name server is running, mail will be configured to use it.
```

In this case, the configuration utility configures MMDF on the host that you are configuring to use the name server automatically.

- If the name server is not configured, you see:

```
The name server is not currently configured on this machine.
Do you plan to configure a name server on the local network? [n]
```

If you do not want to configure a name server, press **<Return>** to continue with the configuration utility.

If you plan to configure a name server, you must do so before configuring MMDf. In this case, enter **"y"**. The configuration utility exits and displays this message:

```
Name server must be configured before mail.
```

Set up the name server and then run **mkdev mmdf** again. For information on configuring the name server tables, refer to Chapter 4, "Configuring the BIND name server," earlier in this guide.

**13. The following screen is displayed:**

```
Many sites do not have complete information about the entire mail network,
but rely on another "smarter" host to determine the correct route that
mail messages should follow to reach their destinations. Any mail that
the local machine is incapable of correctly handling is passed to the
smart host for further processing. In MMDf, this is called the "badhost"
channel.
```

```
Do you have such a "smart" host? [y]
```

If you enter **"n"**, go to the next step; if you enter **"y"** the following screen is displayed:

```
What is its name? (q if you have changed your mind) []
```

**14. The following screen is displayed:**

```
Another option, which is often used on large sites is to have a central
machine which contains complete knowledge about all the users on the site,
and only maintaining local lists on each machine. MMDf provides the
facility to forward mail containing unrecognised local addresses to a
smarter host which will have a complete user data base (via the "baduser"
channel).
```

```
Do you have such a "smart" host? [y]
```

If you enter **"y"** the following screen is displayed:

```
What is its name? (q if you have changed your mind) []
```

15. When you finish, the configuration utility displays information similar to the following as it creates and edits the MMDF configuration files:

```
Creating the mmdftailor file: header, host name info, support
address, alias tables, local domain and channel, SMTP, root
Building the alias tables (mostly empty)
A Mailing list alias table (alias.list) already exists, skipping
A general user alias table (alias.user) already exists, skipping
Building channel files
local, list, SMTP, UUCP, badusers, badhosts
Building the domain tables
local, domain, root
done
building the database
```

The configuration is now complete.

## *Preparing MMDF for use*

---

Before you can use your mail system, you must perform steps to notify the UNIX system of the new configuration. The following sections contain information on restarting **deliver** and making this change permanent in the MMDF system startup file.

### *Restarting the deliver daemon*

When you enter multiuser mode, the system automatically starts a **deliver** process for the local channel only. Each time someone on the system sends or receives a message, the **deliver** daemon runs, placing the message in the correct channel.

With the MMDF configuration utility, you configured MMDF to use additional channels. However, the configuration utility does not affect the **deliver** daemons that are currently running. To make the changes to your MMDF configuration take effect, stop and restart any **deliver** daemons using these commands:

1. As **root**, enter the following command:

```
ps -ummdf | grep deliver
```

This command displays any **deliver** processes running on your system. For example:

```
285 ? 0:00 deliver
```

2. Stop the **deliver** process with **kill(C)**. For example:

```
kill 285
```

3. Now, restart **deliver** with this command:

```
/usr/mmdf/bin/deliver -b
```

This command restarts the **deliver** daemon so that it runs every 10 minutes. If you want **deliver** to run more often, use the **-T** option. We

recommend that you set up **deliver** to run every 60 seconds using the following command:

```
/usr/mmdf/bin/deliver -b -T60
```

**NOTE** When you restart the **deliver** daemon, the change is only temporary. The next time you reboot, the system reads the original **deliver** command in */etc/rc2.d/S86mmdf* and starts **deliver** for the local channel only. To make your changes permanent, modify the *S86mmdf* file using the information in the next section.

## Modifying MMDF system startup

Unless you want to run multiple **deliver**(ADM) daemons, you do not need to modify MMDF system startup. The MMDF system startup file */etc/rc2.d/S86mmdf* already includes information for starting the **deliver**(ADM) daemon on all the configured channels. The system reads the *S86mmdf* file automatically when you enter **init** state 2 (multiuser mode).

By default, the line in this file looks like the following:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b"
```

To edit *S86mmdf*:

1. First, log in as **root**.
2. Edit */etc/rc2.d/S86mmdf*.
3. If you want **deliver** to run more often than the default of every 10 minutes, add the **-T** option on this line. For example, set up your system so that **deliver** runs every 60 seconds using the following **deliver** startup line:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -T60"
```

**NOTE** When you start the **deliver** program with the **-b** option only, *one* **deliver** process manages *all* the channels that you specified with the configuration utility. If you want different **deliver** processes to manage each of your channels or more than one **deliver** process for a single channel, refer to "Running multiple deliver daemons" (page 767).

## Other common configuration tasks

Depending on your configuration, you might need to perform some additional MMDF configuration tasks after running the configuration utility and restarting **deliver**. This section covers some common configuration tasks; for additional configuration information, refer to "Modifying MMDF configuration" (page 759) and "Advanced MMDF configuration tasks" (page 771).

### *Creating aliases for users*

If you chose to hide your local host name behind your domain name in step 6, you should add the login names of the users on the system and machine names where they receive mail to the `/usr/mmdf/table/alias.user` file. The configuration utility creates `alias.user` for you; add information to this file using the following format:

**user:**    `user@machine.domain`

For example, to map *andrei* to the machine *scribe* in the *npr.com* domain, use this format:

**andrei:**   `andrei@scribe.npr.com`

See "Alias files" (page 747) and "Maintaining user aliases" (page 760) for more information on setting up aliases on your system.

### *Converting XENIX alias files*

If you are converting your mail system from XENIX to UNIX, you must use the `mmdfalias` utility to convert a XENIX-style `/usr/lib/mail/aliases` file to the correct format for MMDF. For information on how to do this, see "Converting from XENIX" (page 763).

### *Changing logging levels*

By default, the configuration utility sets the logging levels to log fatal errors only (FAT). If you are switching to MMDF from a `sendmail`-based system, you might want to change the `MMSGLOG` tunable parameter, in the `/usr/mmdf/mmdftailor` file, from FAT to FST to retain a logging level equivalent to "mqueue/syslog". See "Logging levels" (page 746) and "Changing MMDF parameters" (page 784).

## *Rebuilding the MMDF hashed database*

---

The configuration utility rebuilds the MMDF hashed database automatically after modifying the configuration files. However, if you modify the alias files or `/usr/mmdf/mmdftailor` after running the MMDF configuration utility, you must rebuild this database manually.

To do this, enter the following commands as user `mmdf`:

```
cd /usr/mmdf/table
./dbmbuild
```

**NOTE** You must rebuild the MMDF database every time you make changes to the alias or routing files in the `/usr/mmdf/table` directory.

## MMDF configuration files

The configuration utility provided with your MMDF distribution builds configuration files sufficient for most sites, but not for every site; see "Configuration utility limitations" (page 718) for more information. In cases where the files that the configuration utility builds are insufficient, you must edit these files manually. This section describes the different configuration files, how MMDF uses them, and how to modify them to work with your configuration. This section also explains how to test and maintain the MMDF system.

The material presented in this section assumes that you have already set up the channels over which you plan to route mail and that you built the initial MMDF configuration using the configuration utility.

This section begins with an introduction to the different configuration files and formats that MMDF uses to route mail.

Table 14-1 briefly describes the MMDF configuration files.

**Table 14-1 MMDF configuration files**

| File(s)       | Description                                                                                                    |
|---------------|----------------------------------------------------------------------------------------------------------------|
| mmdftailor    | defines all the mail attributes for the local machine                                                          |
| alias files   | defines aliases for mailing lists, programs, users and system accounts                                         |
| domain files  | matches host names to fully qualified host names                                                               |
| channel files | expands fully qualified host names to specify the addressing information necessary to reach the host or domain |

The following sections describe the format of these files in detail. The tables(F) manual page also contains information about the file formats.

### *The mmdftailor file*

The `/usr/mmdf/mmdftailor` file defines all the mail attributes for the local machine, such as its name, the domain, channel, and alias files to use, how to set up each channel, and how to perform logging.

By default, MMDF is distributed with a simple *mmdftailor* file that is configured for local mail only. When you perform the initial MMDF configuration (as described in “Running the configuration utility” (page 727)) the configuration utility modifies the organization of the default *mmdftailor* file. This section describes the *mmdftailor* file generated by the configuration utility. Your *mmdftailor* file might be set up differently (for example, the names of your hosts are different and your *mmdftailor* file might not include all the channels in the example).

Table 14-2 briefly describes the keywords in the *mmdftailor* file; the sections that follow the table explain these keywords in more detail.

**Table 14-2** *mmdftailor* file keywords

| Keyword            | Description                                                                                                                                                                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MLDOMAIN</b>    | defines the top-level domain name (for example, <i>COM</i> or <i>GOV</i> )                                                                                                                                                                          |
| <b>MLNAME</b>      | specifies the host name ( <i>scribe</i> ) or site name ( <i>npr</i> ) for hidden hosts                                                                                                                                                              |
| <b>MLOCMACHINE</b> | specifies the host name (for hidden hosts)                                                                                                                                                                                                          |
| <b>UUname</b>      | specifies the host name for UUCP                                                                                                                                                                                                                    |
| <b>MSUPPORT</b>    | sets the address ( <i>postmaster</i> ) to send undeliverable mail and requests for information                                                                                                                                                      |
| <b>MTBL</b>        | associates an abbreviated name with the more descriptive name for the alias, channel and domain files (also called “tables”). These abbreviated names are then used throughout the <i>mmdftailor</i> file as shorthand to refer to the table files. |
| <b>ALIAS</b>       | defines the various sources for alias information, using the abbreviated names specified in the <b>MTBL</b> definitions                                                                                                                             |
| <b>MCHN</b>        | defines the channels available to MMDF for mail transport*                                                                                                                                                                                          |
| <b>MDMN</b>        | describes the domains known to MMDF                                                                                                                                                                                                                 |
| <b>MMSGLOG</b>     | controls logging information from <b>deliver(ADM)</b> and <b>submit(ADM)</b>                                                                                                                                                                        |
| <b>MCHANLOG</b>    | controls logging that <b>MMSGLOG</b> does not control                                                                                                                                                                                               |
| <b>MLCKTYPE</b>    | specifies the mailbox locking protocol                                                                                                                                                                                                              |

\* MMDF requires at least one channel (*local*), for delivering local mail. See “Local mail configuration” (page 743) for more information.



The **MTBL**, **ALIAS**, **MCHN**, and **MDMN** keywords accept specific parameters; Table 14-3 briefly describes the parameters in the default *mmdf.tailor* file. See the **mmdf.tailor(F)** manual page for complete information about each of these keywords and their associated parameters.

**Table 14-3** *mmdf.tailor* file parameters

| Keyword      | Parameter      | Description                                                                                                                                                          |
|--------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MTBL</b>  | name<br>file   | specifies the shortname (table) for file<br>describes the filename to associate with the shortname                                                                   |
|              | show           | sets the description to display with programs, such as <b>checkup(ADM)</b>                                                                                           |
| <b>ALIAS</b> | table          | specifies the name of the table to associate with the alias entry                                                                                                    |
|              | nobypass       | prevents the <i>~address</i> alias bypass mechanism from working on the aliases in the specified table                                                               |
|              | trusted        | states that any alias in the described file is permitted to deliver mail to files or pipes using the permissions of any user on the system (including <i>root</i> )* |
| <b>MCHN</b>  | show           | sets the description to display with programs                                                                                                                        |
|              | ap             | specifies the type of address parsing used for the header of outgoing messages                                                                                       |
|              | name           | specifies the name of the channel                                                                                                                                    |
|              | mod            | sets the delivery mode for the channel                                                                                                                               |
|              | host           | specifies the name of the host that is being contacted by this channel                                                                                               |
|              | tbl            | defines the file that lists the hosts that are accessible via the specific channel                                                                                   |
|              | confstr<br>pgm | specifies channel-specific configuration parameters<br>defines the channel program to invoke for this channel                                                        |
| <b>MDMN</b>  | show           | sets the description to display with programs                                                                                                                        |
|              | table          | specifies the name of the table containing information that describes the sites in this domain                                                                       |
|              | dmn            | specifies the name of the domain                                                                                                                                     |

\* Normally, because this permits such accessibility, you should only set the *alias* file (used for administrative aliases) to **trusted**.

## MLDOMAIN

The **MLDOMAIN** keyword describes the top-level domain used by the local host. If the local machine has a registered top-level or subdomain name, the top-level domain name for your site, such as *COM* or *EDU*, appears in the **MLDOMAIN** line. If your host does not have a top-level domain, use **UUCP**. (**UUCP** is a generic name that **MMDF** uses for **UUCP** connections.) The **MLDOMAIN** line in the following example shows *COM* as the top-level domain:

```
MLDOMAIN com
```

## MLNAME and MLOCMACHINE

Generally, **MLNAME** describes the name of the local host. In the following example, *scribe* is the host name:

```
MLNAME scribe
```

However, if you are “hiding” the local machine behind a domain name, **MLNAME** describes the subdomain name of your site (for example, *npr*) and **MLOCMACHINE** describes the name of the local host. (**MLOCMACHINE** is only used if you are hiding the local host.)

For example, if you are hiding the local host, *scribe*, behind the *npr* subdomain, these lines look like the following:

```
MLNAME npr
MLOCMACHINE scribe
```

For more information about hiding the host name, see “Hide your host name?” (page 724).

## UUname

The **UUname** line describes the name used with **UUCP**; this definition must exist for **UUCP** to work properly. In most cases, this name is the same as the host name (in either **MLNAME** or **MLOCMACHINE**), but can be different if required. For example:

```
UUname scribe
```

## MSUPPORT

The **MSUPPORT** line describes the address to send mail delivery problem notifications. If your site is connected to the Internet, you must define **MSUPPORT** as *postmaster*. On the Internet, people use this address to send any inquiries about user and host names in the domain. In addition, RFC821/822 (Request for Comments, an Internet standard) requires that every host on the Internet provide the reserved *postmaster* mailbox. Use the following line:

```
MSUPPORT postmaster
```

Note that the address you specify with **MSUPPORT** must be legal; if it is not and MMDF cannot deliver the original undeliverable mail to the support address, MMDF creates a new piece of mail that is undeliverable, and so on until the machine runs out of processes.

You can either create an alias in the *alias.ali* file to redirect *postmaster*'s mail to a user on the system, or create a *postmaster* account. For more information, see "Alias files" (page 747) and "Changing the postmaster alias" (page 760).

## Alias configuration

The **MTBL** alias configuration keywords identify the filenames for the default alias files. By default, these **MTBL** lines look like the following:

```
MTBL name=alias, file="alias.ali", show="Administrative aliases"
MTBL name=lalias, file="alias.list", show="Mailing list aliases"
MTBL name=auser, file="alias.user", show="General user aliases"
```

Each **name** parameter defines the short name for the actual filename containing alias information (specified by **file**). Thus, the first **MTBL** entry identifies "alias" as the shortname for the *alias.ali* file. (MMDF uses this file to define the system administrative aliases for the local host.) See "Alias files" (page 747) for information about these files.

The **ALIAS** alias configuration keywords define additional information about the alias files in the **MTBL** lines. The following example shows the default **ALIAS** lines:

```
ALIAS table=alias, nobypass, trusted
ALIAS table=lalias, nobypass
ALIAS table=auser
```

**NOTE** The order of the alias files defined in this section determines the order that MMDF searches the files to find an alias.

For more information, see "Alias files" (page 747).

## Local mail configuration

As with alias configuration, the **MTBL** keywords identify the filenames that contain local mail configuration information. The following example shows these table definition lines:

```
MTBL name=local, file="local.chn", show="Local Host Aliases"
MTBL name=locdom, file="local.dom", show="Local Domain"
```

In this example, the first **MTBL** entry identifies "local" as the shortname for the *local.chn* file. (The *local.chn* file contains information about the local channel on the local machine.) See "Channel files" (page 753) for more information about these files.

Below the MTBL definitions are the MCHN and MDMN definitions for local mail delivery. By default, the MCHN line for local delivery looks like the following:

```
MCHN local, show="Local Delivery", ap=822, mod=imm
```

The first parameter after MCHN is an arbitrary name that describes the channel (in this case, *local*). For a description of the MCHN parameters, see the *mmdftailor(F)* manual page. MMDF searches the channel tables in the order that the MCHN definitions appear in *mmdftailor*.

The local mail MDMN line looks like the following:

```
MDMN "npr.COM", show="Local domain", table=locdom
```

The first parameter on the MDMN line specifies the name of the domain that the table lists (in this case, the local domain *npr.COM*).

### List processing configuration

The list processing configuration section of *mmdftailor* defines the table and channel that MMDF uses to process mailing lists. By default, these lines look like the following example:

```
MTBL list, file="list.chn", show="List Channel"
MCHN list, show="List Processing", ap=same, mod=imm,
 host="scribe.npr.COM"
```

The MTBL line defines the shortname for the file *list.chn*. This file contains information about passing mail addressed to mailing lists to the list-processor program (see "The alias.list file" (page 748) for more information).

### SMTP configuration

If you configured your system to route mail over TCP/IP, your *mmdftailor* file contains MTBL and MCHN entries like the following:

```
MTBL smtpchn, file=smtp.chn, show="SCO SMTP Channel"
MCHN smtp, show="SCO SMTP Delivery", ap=822, tbl=smtpchn, mod=imm,
 confstr="scribe.npr.COM"
```

MTBL describes the SMTP channel file, *smtp.chn*. MCHN gives more information about the SMTP channel; the *mmdftailor(F)* manual page covers these parameters in more detail.

### Local domain table configuration

The MTBL and MDMN entries in the local domain table configuration section describe the machines in the local domain. These lines look like the following:

```
MTBL domain, file=domain.dom, show="Local Ethernet"
MDMN "npr.COM", show="Local Ethernet", table=domain
```

The first parameter on the MDMN line specifies the name of the domain that the table lists (in this case, the *npr.COM* domain). The *domain.dom* file describes the machines located in the local domain.

## UUCP configuration

If you configure MMDF to route mail over UUCP, the configuration utility adds parameters like the following that describe the UUCP configuration information:

```
MTBL uuchn, file="uucp.chn", show="NPR UUCP Channel"
MTBL uudom, file="uucp.dom", show="NPR UUCP Domain"

MCHN uucp, show="NPR UUCP Delivery", tbl=uuchn, ap=same
MDMN "UUCP", show="UUCP Domain", table=uudom
```

The two MTBL entries define the UUCP channel (*uucp.chn*) and UUCP domain (*uucp.dom*) files. The MCHN line describes more information about the UUCP channel and MDMN defines the file that describes the machine accesses via UUCP.

## The badhosts channel

If you specified a "smart host" to redirect mail destined for machines that the local host does not recognize, the MCHN badhosts line looks like this:

```
MCHN badhosts, show="Last-Chance Routing", pgm=smtp, tbl=smtpchn,
ap=822, host="rocky.npr.COM", confstr=scribe.npr.COM
```

The MCHN host parameter describes the smart host (in this case, *rocky.npr.COM*). In this case, the local host accesses the smart host via SMTP. The pgm parameter specifies the filename of the channel program (*smtp*) in the */usr/mmdf/chans* directory and tbl specifies the file (*smtp.chn*) that describes the machines that the host accesses with SMTP.

The badhosts is not really a channel because it is not associated with its own transport program. In this example, the pseudo-channel uses the SMTP channel to relay mail to a more intelligent host. If the badhosts channel does not exist, MMDF returns mail to an unknown host to the sender.

## The badusers channel

If you specified a "smart host" to redirect mail to users that the local host does not recognize, the MCHN badusers line looks like this:

```
MCHN badusers, show="Last-Chance Routing", pgm=smtp, tbl=smtpchn,
ap=822, host="moose.npr.COM", confstr=scribe.npr.COM
```

The MCHN host parameter describes the smart host (in this case, *moose.npr.COM*). See the previous section for information about the pgm and tbl parameters.

## *The root domain table*

The **MTBL** line defines the *root.dom* file which contains any domain information not named in other domain files.

```
MTBL rootdom, file="root.dom", show="Root Domain"
MDMN "", show="Root Domain", table=rootdom
```

The **MDMN** entry defines the name of root domain. (The root domain definition has no name ("") because the root domain file (*root.dom*) can contain entries for many different domains.)

## *Logging levels*

By default, the configuration script sets **MMSGLOG** and **MCHANLOG** to the **FAT** logging level; this level logs fatal errors only. These entries look like the following:

```
MMSGLOG level=FAT
MCHANLOG level=FAT
```

You can change these logging parameters to a higher level, such as **FST** to log full statistics, or add different logging controls, such as **AUTHLOG** to control authorization information. See the section on **MCHANLOG** in the **mmdftailor(F)** manual page for a list of levels. "Changing MMDF parameters" (page 784) also contains information about these logging levels. For more information about the MMDF logging files, see the **logs(F)** manual page.

## *Mailbox locking style*

By default, MMDF uses the standard System V **fcntl()** kernel file locking protocol to lock users' mailboxes. However, if users on the system use MUAs that do not use the default locking protocol, you can configure the locking type with the **MLCKTYPE** keyword. The default **MLCKTYPE** line looks like the following:

```
MLCKTYPE advisory
```

See "Changing MMDF parameters" (page 784) for more information on setting locking protocols.

## Alias files

An *alias* is an abbreviated name that MMDF translates into a larger string (a mail address or list of addresses). Aliases are useful for specifying a single name to represent a group of users. You might want to create an alias called *sales* to represent all the members of the Sales department in the company. To do this, you define an alias file in the */usr/mmdf/mmdftailor* file and then create the alias file in */usr/mmdf/table*. In the alias file, the *sales* alias entry might look like the following:<sup>4</sup>

```
sales: joe, jane, bob, mike, karen, ann
```

When users want to send mail to everybody in the Sales department, they can use the *sales* alias like this:

```
mail sales
```

This sends a copy of the message to everyone in the Sales department.

You can name your alias files anything you like; however, you must define the file names in the */usr/mmdf/mmdftailor* file (see “The mmdftailor file” (page 739) for details). By default, MMDF provides the files listed in Table 14-4 that you can use to specify aliases for user names:

**Table 14-4** MMDF alias files

| File       | Description                                                                                                                                                    |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| alias.list | aliases for lists of users                                                                                                                                     |
| alias.user | aliases mapping users to their “home” host machines (the machines on which they receive their mail) and nicknames for local users                              |
| alias.ali  | local machine system administrative aliases, programs, or files. At the minimum, this file should identify the aliases for <i>mmdf</i> and <i>postmaster</i> . |

These files are located in the */usr/mmdf/table* directory.

\* The colon (:) following the alias name is optional.

**NOTE** You should *not* create aliases for remote users (users not located at the local site) in the alias files. In other words, do not create an alias like the following, where *moocow.uucp* is not a local machine at your site:

```
david: david@moocow.uucp
```

In this case, if a remote user sends mail to *david@npr.com*, the message is delivered to *david@moocow.uucp*.

In general, users should set private remote-user aliases using their MUA (for example, use the `alias` command in `$HOME/.mailrc`). For more information, about creating private aliases, see the chapter on electronic mail in the *User's Guide*.

## The *alias.list* file

Use the *alias.list* file to create multiple-user aliases. With list aliases, MMDF processes the mail using the `list` channel and it appears as being from the sender instead of from the first person in the "To:" line.

To use the `list` channel, you must specify the following three lines for each alias:

- the name of the list
- the "*name-outbound*" line that contains the logins that comprise the list
- the "*name-request*" line that provides a login of the list maintainer who makes additions and deletions to and from the list

The following example shows how to use these three lines:

```
writers: writers-outbound@list-processor
writers-outbound: hanna, dianna, george, laurie, meg, naomi, steve
writers-request: hanna
sales: sales-outbound@list-processor
sales-outbound: joe, jane, mike, karen, ann, uksales
sales-request: joe
```

The `list-processor` keyword is a reserved word in MMDF and indicates access to the `list` channel for processing mailing lists. The *sales-request* alias provides a way to request additions or deletions to the *sales* alias.

When defining an alias that contains many user names, you can use a backslash character (`\`) as a line-continuation character. Use quotation marks (`"`) to delimit a string containing spaces or punctuation. When using an alias to define another alias, be careful not to create an alias loop.



You can also use the output redirection symbol (>) with the pipe character (|) to do more complex processing such as redirecting messages to files. For example:

```
Loguucp: "network//usr/spool/log/uucp"
Logmlog: "network|cat -v >>/usr/spool/log/mlog"
printer2: "network|/usr/bin/lp -dprinter2"
```

In this example, MMDF pipes mail addressed to *Logmlog* to the *cat(C)* command to log the mail in the *mlog* file. MMDF pipes mail addressed to *printer2* to the *lp(C)* command for printing. These redirection alias examples use the user and group IDs of the user *network*. Although *network* is appropriate in most cases, you can specify any user named in the */etc/passwd* file on your system.

If you have a long list of names for an alias, you might want to include them in another file (instead of listing them directly in the *alias.list* file). To do this, use the *:include* keyword. For example:

```
staff: staff-outbound@list-processor
staff-outbound: ":include:/etc/alias/staff"
staff-request: ross
```

In this example, the *:include:* line specifies that you want to use the names listed in the */etc/alias/staff* file to define the alias. Note that you cannot use the backslash character (\) as a line-continuation character for lists of names in an *:include* alias list.

For details on setting up mailing lists, see the *list(ADM)* manual page.

## The *alias.user* file

To map users to specific machines, specify aliases in the *alias.user* file. Use this file when you want each person at your site to receive mail on a particular machine. The following example shows how to set up *alias.user*:

```
andrei: andrei@scribe
george: george@dera
hanna: hanna@scribe
karen: karen@guardian
```

## The *alias.ali* file

Use the *alias.ali* file for aliases that are not specific to *alias.list* or *alias.user*. This file usually contains aliases related to system administration, such as:

```
root: david
mmdf: david
postmaster: david
uucp: david
```

In this example, all mail addressed to *root*, *mmdf*, *postmaster*, and *uucp* is redirected to *david* on the local machine.

In general, do **not** create aliases for remote users (users not located at the local site) in the *alias.ali* file.

### *How MMDF uses alias files*

When mail is addressed to *postmaster*, MMDF routes the mail by first searching the hashed alias table from *alias.ali* to expand the *postmaster* alias to the associated user name. For example, the *postmaster* entry in the *alias.ali* file:

```
postmaster: david
```

Then, MMDF searches the *alias.user* file to find the local machine name associated with the user name. For example, the *alias.user* file contains an entry like this:

```
david: david@golem
```

MMDF then uses this information when searching the various *.dom* files, which map the local machine name to a fully qualified host name and the *.chn* files, which map the fully qualified host name to information on how to route the message. "How MMDF routes mail" (page 755) explains the process for searching configuration files further.

## *Domain files*

---

The domain files are used to match a host name to its fully qualified host name. Domain files serve two purposes: to convey information to MMDF about how machines are connected, and to specify special routing considerations for subdomains or top-level domains.

Domain files are named for the domain that they describe (except for *root.dom*, which contains domain information not named in other domain files); the filenames end in *.dom*. The general practice is to have a separate domain file for the domain in which the host machine resides.

The operating system distribution includes four domain files in the */usr/mmdf/table* directory. Table 14-5 lists these files.

**Table 14-5** MMDF domain files

| Domain | Domain file | Describes                                |
|--------|-------------|------------------------------------------|
| local  | local.dom   | local machine                            |
| domain | domain.dom  | machines in the local domain             |
| uucp   | uucp.dom    | machines in UUCP domain                  |
| root   | root.dom*   | domains not listed in other domain files |

\* The *root.dom* file also contains information about how to access top-level domains, such as *MIL* and *GOV*. (The name *root* implies "top-level", as in a hierarchy.)

You can create new domain files for each domain. For example, create a domain file for the *npr.COM* domain and name it *npr.dom*. You do not need to use the *\*.dom* naming scheme; however, this filename extension makes it easier to determine the purpose of the file.

## Domain file format

Each domain file consists of two columns of information: the left column lists the host name and the right column lists the fully qualified name for that host. The domain names can be either upper- or lowercase. Use tabs, spaces, a colon, or a combination of these characters to separate the first column from the second.

The name of a domain file determines the domain names for which MMDF searches. For instance, the domain file for UUCP generally contains entries for names in the following form:

```
machine: machine.UUCP
```

However, you can create an entry to map a specific UUCP address to another address. For example, to map *research.UUCP* to *research.jcn.com*, the entry looks like the following:

```
research: research.jcn.com
```

The following four sections give examples of different domain files.

## The local.dom file

The *local.dom* file describes the local host. For example, the contents of the *local.dom* file on the host *scribe* in the domain *npr.COM* look like the following:

```
scribe: scribe.npr.COM
```

## The domain.dom file

The *domain.dom* file describes the machines in the local domain (independent of the channel that MMDF uses to reach each machine). For example, if there are four machines in the domain *npr.COM*, the *domain.dom* file looks like the following:

```
scribe: scribe.npr.COM
huey: huey.npr.COM
dewey: dewey.npr.COM
louie: louie.npr.COM
```

Note that the *domain.dom* file maps each machine to the fully qualified host name in the *npr.COM* domain.

### *The uucp.dom file*

The *uucp.dom* file specifies the hosts in the UUCP domain. For example, if your host connects to the remote machine *cactus* via UUCP, the *uucp.dom* file looks like the following:

```
cactus: cactus.UUCP
palm: palm.UUCP
```

In this case, MMDF directs any mail sent to the *cactus* or *palm* machines to the UUCP network.

### *The root.dom file*

The *root.dom* file defines the hosts and domains not defined in the other domain files. For example, if your host connects to the UUnet network system, you can set up MMDF to send all mail to specific domains to *uunet.UU.NET*. To do this, set up your *root.dom* file like the following:

```
COM: uunet.UU.NET
EDU: ucsc.EDU
MIL: uunet.UU.NET
GOV: star.GOV
NET: uunet.UU.NET
```

In this example, all mail directed to the *COM*, *MIL* is sent out on UUnet; mail to *GOV* goes to *star.GOV*; and mail sent to the *EDU* domain goes to *ucsc.EDU*.

### *LAN considerations*

If you are configuring MMDF for use on a local area network (LAN), you can use the domain files to distribute the processing load on the machines, or you can designate a special mail server machine to route all the messages.

You have the following choices:

- **MMDF Server** — Designate one machine as the network server. (This machine may also have outer world access.)

Each machine's domain file only needs to include **badhosts** and **badusers** channels that contain the fully qualified host name of the server. The domain file can be the same for each machine in the network. The network can grow, and machines can be added and removed with no effect on the domain files of other computers. The disadvantage is that the server receives a great deal of traffic and should be dedicated to its task. In addition, if the server is down, so is all electronic mail between machines. The best policy is a system of machines grouped around a server that has knowledge of another server; each server has knowledge of yet another and so on.

- **Distributed Processing** — Give each machine's domain file knowledge of each other machine in the network.

The advantage is that networked machines can operate independently of each other. One machine's crash has no effect on the mail capability of the others (unless that machine is connected to another network via UUCP). The disadvantages are that system administration gets geometrically more difficult as you add or remove machines to or from the network. When a machine or user is added to or removed from the network, you must update all domain or alias files to recognize the change. Because a domain file can contain redundant information about the local machine, you can use one domain file on every machine. You should only use distributed processing for small networks.

You can greatly simplify the distributed processing configuration by running a name server program on your network. For details on setting up the name server, see Chapter 4, "Configuring the BIND name server," earlier in this guide.

- **Gateways** — Use a gateway to connect a network to another Local Area or Wide Area network. (This setup is derived from the server setup.) In this case, the server machine is typically connected to more than one network.

In addition to containing the names of the local machines on the LAN, a gateway machine also contains the names of the other machines to reach over the other networks. This information is kept in the respective domain and channel files for the other networks and also in the *root.dom* file on the gateway machine. Other machines on the LAN use the **badhosts** channel to route non-local mail to the gateway machine, or a *root.dom* file that lists all the top-level domains as routing through the gateway. To avoid overloading the gateway machine, the other machines on the LAN use local domain files as described in the earlier "Distributed Processing" bullet.

## Channel files

MMDF uses the channel files to determine the channel to use for outgoing mail and the address of the host on that channel. Channel files map the fully qualified host name (as determined from the domain file entries) to channel-specific addressing information. For example, the UUCP channel file maps host names to UUCP paths (using exclamation points) specifying how to get to each host.

The operating system distribution includes four channel files in the */usr/mmdf/table* directory. Table 14-6 lists these files.

**Table 14-6** MMDF channel files

| Channel | Channel file | Describes                  |
|---------|--------------|----------------------------|
| local   | local.chn    | local machine              |
| list    | list.chn     | list-processor             |
| smtp    | smtp.chn     | machines accessed via SMTP |
| uucp    | uucp.chn     | machines accessed via UUCP |

The MCHN definitions in the */usr/mmdf/mmdftailor* file direct MMDF to search the specified *.chn* files in the */usr/mmdf/table* directory for channel definitions.

Separate the left and right columns in the channel files by a space or tab, a colon character ":" or both.

### *The local.chn file*

The *local.chn* file contains entries describing all the names local host is called, mapping them to the local host name. For example, if the local host is *scribe.npr.COM*, then the local host is known as *scribe* on the local machine. The *local.chn* file maps the local host name (on the right) to the different ways people might refer to *scribe*. This file looks like the following:

```
scribe: scribe
npr.COM: scribe
scribe.npr.COM: scribe
```

### *The list.chn file*

The *list.chn* file contains information about the list-processor program:

```
list-processor: list-processor
list-proc: list-processor
```

The left column is a pseudo-host defined in a mailing list alias (see "The alias.list file" (page 748) for details). These entries tell MMDF to pass mail addressed to a mailing list to the list-processor program.

### *The uucp.chn file*

The *uucp.chn* file contains entries describing the hosts that your host connects to using the UUCP channel and how to route mail to those hosts. For example, the format of this file looks like the following:

```
mcvax.UUCP: uunet!mcvax!%s
sri-nic.ARPA: uunet!sri-nic.arpa!%s
uunet.uu.NET: uunet!%s
```

The left column contains the UUCP host name from the domain tables; the right column describes the UUCP address that MMDF uses to direct mail to that host. The "%s" at the end of the UUCP address means to use the rest of the address from this point on. In other words, when mail is addressed to the user *hillis* at *mcvax.UUCP*, the UUCP channel passes the mail to UUnet along

with the rest of the UUCP address (*mcvax!hillis*). The second entry in this example shows how a domain name (*sri-nic.ARPA*) can be used within a UUCP path.

Channel file entries for the UUCP channel (in the *uucp.chn* file) when the destination machine is multiple hops away, appear as follows:

```
stooges.UUCP: moe!curly!larry!stooges!%s
```

Specify the **address** of the host on the right hand side, where the **address** is a UUCP path.

### *The smtp.chn file*

The *smtp.chn* file describes the hosts that you connect to using TCP/IP and the IP addresses of those hosts. The format of this file looks like the following:

```
cocoa.npr.COM: 123.456.789.1
caramel.npr.COM: 123.456.789.2
taffy.npr.COM: 123.456.789.3
```

The left column contains the fully qualified host names for the hosts that you connect to with TCP/IP and the right column contains the IP addresses. For more information about configuring MMDF to route mail over TCP/IP, see the *TCP/IP User's and Administrator's Guide*.

Following the pattern of mapping the host name in the left column to the addressing information for delivering to that host in the right column, you can create channel files for each MCHN definition in *mmdftailor*. The exception to this is the **badhosts** pseudo-channel definition; the **badhosts** channel program is determined at configuration time automatically. However, you can set up your channel files to indicate the channel that you use to reach the "smart host".

## *How MMDF routes mail*

---

This section describes how MMDF uses the information in the configuration files to route mail on your system. Note that MMDF never searches the alias, channel, and domain files directly. When you build the hashed database with **dbmbuild(ADM)**, the contents of these files are stored in tables in the *dbm* database. MMDF then uses the information in this database to route mail.

Mail arrives at and leaves computers using one of several different methods (such as UUCP, or TCP/IP called "channels". The MMDF **submit(ADM)** command accepts the incoming mail from a channel and determines the correct outgoing channel to use based on the destination host. The **submit** program uses the information in the domain tables to map the way the incoming mail describes the destination host to the way the host recognizes that destination

host. Based on the host description, **submit** uses the channel tables to determine the outgoing channel to use to route the message and places the message in the appropriate queue.

Then, using the information from */usr/mmdf/mmdftailor*, **deliver**(ADM) moves the mail from the queue to the appropriate channel. For example, if **submit** places a message in the UUCP channel queue, **deliver** moves that message to the UUCP channel. The **deliver** command can also place mail in a channel and let another program (such as **uux**(C) for the UUCP channel) carry out additional steps to resolve the circumstances dictated by the type of channel. The channel program sends the mail across the network to the proper destination.

### *Searching MMDF domain tables*

---

The **submit**(ADM) command uses the domain tables for two purposes: to specify the fully qualified host name and (optionally) to specify the route to a host by listing the fully qualified host names of one or more intermediate hosts through which mail is to be routed.

First, **submit** separates the fully qualified host name into two parts: the name of the domain table and the host name to match on the left-hand side (LHS) of the entries in the domain table. For example, in the address *david@engr.canada.COM*, the name of the domain table to search is *canada.COM* and the host name to search for is *engr*.

MMDF tests an address for matches against the domain names in the MDMN entries in *mmdftailor*. For example, the address *david@engr.canada.COM* matches the following MDMN entry:

```
MDMN "canada.com", show="Canada Delivery", tbl=canadadom
```

Then, MMDF searches *mmdftailor* for the MTBL entry for *canadadom*:

```
MTBL canadadom, file="canada.dom", show "Canada Delivery"
```

MMDF uses the file associated with *canadadom* (*canada.dom*).

Thus, to route our message to *david@engr.canada.COM*, **submit** uses the following algorithm to search the domain tables for a match in the address:

1. Searches for the host name (*engr*) in the LHS of the domain table.
2. Searches in the LHS of the domain table (in this case, *COM*), if it exists, for the domain name (*engr.canada*).
3. Searches the LHS of the *root* domain table for the fully qualified host name (*eng.canada.COM*).



4. Searches the relevant tables that include *flags=route*<sup>5</sup> in the MTBL line in *mmdftailor* for substrings of the address. For example, search the COM table for *canada*; search the *root* table for *canada.COM* or *COM*.
5. Searches the tables that include *flags=partial*<sup>6</sup> in the MTBL line in *mmdftailor* for the input name, in this case, the fully qualified host name (*eng.canada.COM*), regardless of the domain name.

If **submit** finds no match at all, as a last resort, it uses the **badhosts** channel, if it exists. Because MMDF uses the first domain that has an exact match without looking for other matches in later tables, the order in which you list MDMN definitions is significant. Make sure the local domain table appears first and the *root* domain table is last in *mmdftailor*.

If the address provided is *username@host* (for example, *david@engr*), MMDF performs the same search as described above except that there is no *canada.COM* to match in the MDMN entries in *mmdftailor*. In this case, MMDF searches the *root* domain table and then the tables that include *flags=partial* in the MDMN line.

## Delivery channel programs

A channel is a compiled program that permits a machine to talk to a single type of network communications protocol. Some simple channels only store mail for further processing. Channel programs reside in the */usr/mmdf/chans* directory.

Channel programs handle the communications protocol so that neither the operating system nor the rest of the MMDF system has to know about the intricacies of a particular communication protocol. This handling of the protocol is one of the advantages of MMDF; having specific protocol program modules permits a site to upgrade to other network types without having to rewrite the mail delivery system.

\* The *flags=route* parameter in the MTBL entry enables that table entry to match addresses for an entire subdomain, acting as a gateway. Do not use *flags=route* on tables other than the *root* domain table unless you have internal subdomains.

† Use the *flags=partial* parameter on the *local* domain table so that users do not have to specify the full domain to send mail on the local machine.

The different channels are:

- badhosts** Called when a specified machine is unknown to the local machine. The use of “bad” is really a misnomer; any mail to an unknown machine is sent on this channel. You can assign **badhosts** in the *mmdftailor* file to a channel type such as **uucp** or **smtp**.
- badusers** Called when mail arrives at the local machine, but the user does not have a login on this machine, nor is there an alias for the user on the local machine. Generally, MMDF queues up this mail for submission to another machine with a larger list of users (a “smart host”).
- list** Called to remail messages. This channel simply invokes **submit** and feeds the addresses and text back into the MMDF mail system. This is often used to avoid long address validation or to force the validation to occur in the background for very large mailing lists. This also ensures that MMDF sends any problem reports to the list maintainer.
- local** Called to deliver mail to mailboxes and processes on the local machine.
- smtp** Called to deliver or accept mail from a TCP/IP network connection. The **smtp** channel transfers messages by establishing a TCP/IP connection to a remote machine, and using the Simple Mail Transfer Protocol (SMTP) to send one or more mail messages. The Internet Protocol (IP) allows many local- or wide-area networks to be interconnected transparently. This permits the MMDF SMTP channel to exchange messages with any machine on any network to which it is connected. For example, if your machine connects to the Internet, you can exchange messages directly with any machine in the world that is also connected to the Internet.
- uucp** Called to direct mail to UUCP delivery to another machine, or to accept mail from a UUCP connection from another machine. Incoming mail is converted into the format specified by the Internet technical bulletin, RFC822, available from the InterNIC Registration Services, or InterNIC.<sup>7</sup> Outgoing mail includes a “From<space>” line and the mail path arguments are separated by UUCP exclamation point characters (!).

For more information about UUCP, see Chapter 13, “Building a remote network with UUCP”, of this guide.

Channels act not only as protocol handlers, but in some cases actually initiate the communications to the network or to another machine as needed.

---

\* See “Registering domain names” (page 723) for more information.

For each channel, MMDF provides two programs: an input program and an output program. A channel program is associated with outgoing mail; a server program is associated with incoming mail. For example, the UUCP channel has `/usr/bin/rmail` as its input program and `/usr/mmdf/chans/uucp` as its output program.

On the MMDF system, the server is a channel program that monitors the network for incoming mail. On systems other than MMDF, the function of a mail server is built into the mail delivery system. Figure 14-3 describes the relationship between a channel and server.

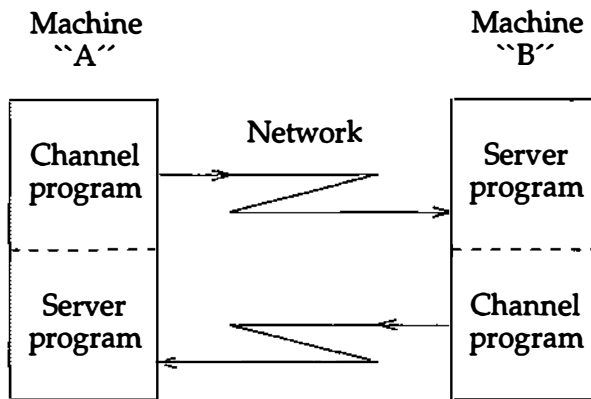


Figure 14-3 Channel and server relationship

Even if MMDF is only being run on one side, the other side, while not correctly called a channel program, performs the same function.

## Modifying MMDF configuration

When you perform the initial MMDF configuration, the configuration utility sets up the appropriate *alias.\**, *\*.dom*, *\*.chn*, and *mmdftailor* files. This initial configuration is sufficient for most sites; if your site configuration changes, you can rerun the configuration utility to update the MMDF files.

However, if this initial configuration does not take care of your site configuration needs, you must edit the MMDF configuration files manually after running the configuration utility. You might also need to edit your configuration files manually if your site configuration changes so that you can no longer configure MMDF with the configuration utility. See "Configuration utility limitations" (page 718) for more information.

## Guidelines for manual configuration

---

Keep in mind the following guidelines when you modify your MMDf configuration files:

- Always perform configuration as *mmdf*.
- Always rebuild the hashed database with **dbmbuild**(ADM) after modifying files. See “Rebuilding the MMDf hashed database” (page 769) for more information.
- If you make any changes to the channel configuration (for example, to add or remove a channel), restart the **deliver**(ADM) daemon.

## Changing the postmaster alias

---

Because RFC821/822 requires that every host provide the special postmaster mailbox, you should never change the **MSUPPORT** line in */usr/mmdf/mmdftailor* to anything other than *postmaster*. However, if the person responsible for mail administration at your site changes, you should change the *postmaster* alias in the */usr/mmdf/table/alias.ali* file.

For example, to change the *postmaster* alias to *hanna*, change the line in *alias.ali* to look like this:

```
postmaster: hanna
```

**NOTE** If you do not define an alias (in the *alias.ali* file) that maps *postmaster* to a real user on the system, you must create a *postmaster* account (using the **sysadmsh**(ADM) command) to direct mail inquiries and problems.

You can designate anyone to receive undeliverable mail, but a local user is best because the address is simpler and, therefore, more likely to be a valid address. Also, with a local user, delivery of the message does not depend on a network connection that might be malfunctioning and therefore responsible for the original mail problem.

## Maintaining user aliases

---

As you add new users to your system, you should update your */usr/mmdf/table/alias.user* file to map the new users to their home machines. For example, if you add *laurie* to the machine *poet* at your site, add the following line to *alias.user*:

```
laurie: laurie@poet
```

You can change the aliases in this file for other reasons, such as when you switch a user to another machine or the user leaves the company. For example, if a user leaves the company, you can redirect their mail to another person, such as their manager. To do this, place a line like the following in *alias.table*:

```
matt: mark@scribe
```

If you create multiple-user aliases (mailing lists), you might need to update the information in your */usr/mmdf/table/alias.list* file periodically. For example, if you add a new alias called *docstyle*, add the following lines to *alias.list*:

```
docstyle: docstyle-outbound@list-processor
docstyle-outbound: joan,kelly,hanna,laurie,tammy,meg,teresa
docstyle-request: joan
```

## Changing the host name

When you install your operating system, the installation procedure prompts you to provide a name for your machine. This machine name is also called the HOST NAME or SYSTEM NAME.

If you decide to change your host name after installing the operating system, use the following procedure:

1. Log in as *root*.
2. Change the host name using the **uname(C)** utility. Use the following command, where *new\_name* is the new name of your host:

```
uname -S new_name
```

For example, to change a machine name to *elgrande*, enter the following command:

```
uname -S elgrande
```

For more information on the **uname** command, see the **uname(C)** manual page.

3. Now, log in as *mmdf* and verify that you are in the */usr/mmdf* directory.
4. Edit the *mmdftailor* file and change all occurrences of the old host name to *new\_name*.
5. Change to the */usr/mmdf/table* directory and edit any *\*.dom* and *\*.chn* files, changing all instances of the old host name to *new\_name*.
6. Now, verify that you are in the */usr/mmdf/table* directory and rebuild the hashed database using the following command:

```
./dbmbuild
```

## 14. Setting up electronic mail

7. Now, log in as *root*, and bring the system down to single-user (system maintenance) mode with the following command:  
**init 1**
8. At the prompt, enter **<Ctrl>d** to bring the system back up to multiuser mode.
9. If you are routing mail over TCP/IP, verify that the */etc/hosts* file contains the new host name.

Now, MMDF uses the new host name.

### *Hiding your host name*

---

If, after running the initial MMDF configuration with the configuration utility, you register your top-level domain or subdomain with the InterNIC, you might decide to hide the host name behind the domain name. In this case, people outside your organization can send mail to people at your site without knowing the host name where they receive their mail.

To hide your host name, you must edit the */usr/mmdf/mmdftailor* file and change the **MLDOMAIN**, **MLNAME**, and **MLOCMACHINE** keywords. For example, if you are not currently hiding the host name, these lines look like the following:

```
MLDOMAIN npr.COM
MLNAME scribe
```

To change the configuration to hide *scribe* behind the *npr* domain name, add **MLOCMACHINE** and change these parameters so they look like this:

```
MLDOMAIN com
MLNAME npr
MLOCMACHINE scribe
```

Now, people can send mail to *andrei@scribe.npr.com* using the *andrei@npr.com* address (they no longer have to specify the host name, *scribe*).

### *Changing your smart host*

---

If your configuration changes so that you no longer rely on a smart host for complete information about the network, or your smart host changes, you must modify the **MCHN** lines for **badhosts** and **badusers** in *mmdftailor*. These lines look something like the following:

```
MCHN badhosts, show="Last-Chance Routing", pgm=smtp, tbl=smtpchn,
ap=822, host="rocky.npr.COM", confstr=scribe.npr.COM
MCHN badusers, show="Last-Chance Routing", pgm=smtp, tbl=smtpchn,
ap=822, host="moose.npr.COM", confstr=scribe.npr.COM
```

In this case, change the value of the **host** parameter (in this example, *rocky.npr.COM* and *moose.npr.COM*) in each **MCHN** definition to the name of the new smart host. For more information about smart hosts, see “Hide your host name?” (page 724).

## Changing error logging levels

---

By default, the MMDF configuration script sets logging levels with the **MMSGLOG** and **MCHANLOG** keywords in the *mmdftailor* file. These keywords are set to the **FAT** logging level (to log fatal errors only) and look like the following:

```
MMSGLOG level=FAT
MCHANLOG level=FAT
```

You can change these logging levels to a higher level, such as **FST**. In addition, you can use the **AUTHLOG** keyword to control authorization information. For a complete list of available logging levels, see the **MCHANLOG** section in *mmdftailor(F)* manual page.

## Converting from XENIX

---

If you are converting from a XENIX to UNIX system or you have a mixed UNIX/XENIX system network, you must convert your XENIX aliases file and your UUCP routing files (as appropriate). (If you install your UUCP configuration files before running the MMDF configuration utility, you only have to convert the alias files.) This section explains how to convert these files.

Before converting your XENIX files, you must install and configure UUCP. You must install and configure these packages so that the files that you use to create MMDF alias, channel, and domain files contain the correct information.

If you did not install the entire distribution, you should install the UUCP and MAIL packages using the **custom(ADM)** utility. See your *Installation and Upgrade Guide* for information on installing packages with **custom**. For information on configuring UUCP, see Chapter 13, “Building a remote network with UUCP”, earlier in this guide.

MMDF provides the utilities listed in Table 14-7 in the */usr/mmdf/table/tools* directory for converting files from XENIX to MMDF.

**Table 14-7** MMDF conversion utilities

| Utility          | Converts this XENIX file:          | To these MMDF files:                                                               |
|------------------|------------------------------------|------------------------------------------------------------------------------------|
| <b>mmdfalias</b> | <code>/usr/lib/mail/aliases</code> | <code>/usr/mmdf/table/alias.list</code><br><code>/usr/mmdf/table/alias.user</code> |
| <b>uulist</b>    | <code>/usr/lib/uucp/Systems</code> | <code>/usr/mmdf/table/uucp.chn</code><br><code>/usr/mmdf/table/uucp.dom</code>     |

For more information, see the **mmdfalias**(ADM) and **uulist**(ADM) manual pages. Note that **uulist** uses **uname**(C) to retrieve the list of accessible sites.

Use the following procedure to convert the XENIX files:

1. First, use the following command to make sure *aliases* and *top* are readable by *mmdf*:

```
ls -l /usr/lib/mail/aliases /usr/lib/mail/top
```

If the *mmdf* user does not have permission to read these files, use **chmod**(C) to change the permissions. Log in as *root* and enter the following command:

```
chmod +r /usr/lib/mail/aliases /usr/lib/mail/top
```

2. Now, log in as *mmdf* and enter the following command:  
**cd /usr/mmdf/table**
3. Use the following commands to convert the routing or alias files, as appropriate:

- To convert alias files, enter:

```
tools/mmdfalias
```

The **mmdfalias** utility creates the files *alias.list* and *alias.user* in the */usr/mmdf/table* directory.

- To convert UUCP routing files, enter:

```
tools/uulist
```

The **uulist** utility creates the files *uucp.chn* and *uucp.dom* in the */usr/mmdf/table* directory.

If any of the MMDF output files already exist, the MMDF utility backs up the original *filename* to *filename-.*



- Now, make sure that the permissions on each of the output files are set to 644, and that the owner and group are *mmdf*. Use the following commands:

```
cd /usr/mmdf/table
```

```
ls -l alias.list alias.user uucp.chn uucp.dom
```

The permissions should look something like the following:

```
-rw-r--r-- 1 mmdf mmdf 150 Jun 11 1992 alias.list
-rw-r--r-- 1 mmdf mmdf 120 Jun 11 1992 alias.user
-rw-r--r-- 1 mmdf mmdf 86 Apr 17 14:23 uucp.chn
-rw-r--r-- 1 mmdf mmdf 61 Apr 17 14:23 uucp.dom
```

If the permissions for any of these files are wrong, use **chmod(C)** to modify them.

Now, all the XENIX routing and alias files are converted to MMDF.

If you are converting to MMDF from XENIX or another UNIX system, the system mailboxes for the users on the system are in the XENIX-style (older UNIX system) format. XENIX format uses the "From<space>" lines to delimit between messages in mail folders; MMDF uses (Ctrl)a characters. You should convert the system mailboxes to use the new MMDF format using the **cnvtmbox(ADM)** utility.

Use the following procedure to do this:

- First, verify that users on the system are not accessing the system mailboxes.
- Log in as *root*. (You must have read permission on the mailbox that you are converting.)
- Now, change to the system mail spool directory (generally, */usr/spool/mail*).
- For each system mailbox in the mail directory, run the following commands:

```
/usr/mmdf/bin/cnvtmbox old_mailbox new_mailbox
mv new_mailbox old_mailbox
```

By default, the **mailx(C)** program maintains the current format of every mail folder that users read with **mailx**. For example, if a mail folder is in MMDF format (with (Ctrl)a delimiters), **mailx** uses this format when adding messages to the folder. If a user runs **mailx** with a folder that has not been converted to MMDF format, it prompts the user to convert the folder.

If any users on the system use an MUA other than **mailx**, advise these users to run **cnvtmbox** on each of their mail folders. As long as users have read permission on a mail folder and write permission on the output, they can convert it to the MMDF format with **cnvtmbox**.

## Modifying UUCP host configuration

---

If you change your UUCP host configuration, use the information in this section to update your MMDF configuration files.

### Specifying options to *uux*

If you use UUCP to route mail, you can specify the options that *uux*(C) should invoke when sending files and executing commands on the remote machine. By default, MMDF uses the following options to *uux*:

```
uux - -r
```

This setting specifies that UUCP queue the job until the next time *uucico*(ADM) runs rather than transfer it immediately (see *uucico*(ADM) for more information).

If you want to change this, add the *UUXSTR* keyword to the */usr/mmdf/mmdftailor* file. For example, if you want *uucico* to run immediately, the following *UUXSTR* setting is correct:

```
UUXSTR 'uux -'
```

In general, we recommend that you use one of the following *UUXSTR* settings:

```
UUXSTR 'uux -'
```

```
UUXSTR 'uux - -r'
```

```
UUXSTR 'uux - -r -gA'
```

For more information on these options, see the *uux*(C) manual page.

### Adding or removing a UUCP host

To add a host that you access via UUCP to your system, use the following procedure:

1. Add the UUCP path to route mail to the host to the */usr/mmdf/table/uucp.chn* file. For example, to add the UUCP host *dudley*, add the following line to *uucp.chn* on the local host:

```
dudley.UUCP: dudley!%s
```

The "%s" specifies to use the remainder of the address indicated in the "To:" mail header from this point.

2. Add the UUCP host name to the *uucp.dom* file. For example:

```
dudley: dudley.UUCP
```

3. Now, rebuild the hashed database with **dbmbuild**.

To remove a UUCP host, remove the entries for that host from the *uucp.chn* and *uucp.dom* files and rebuild the hashed database.

## Adding or removing an SMTP host

Use the following procedure to add an SMTP host to your MMDF configuration:

1. For each new host in */etc/hosts*, add the fully qualified host name and IP address to the */usr/mmdf/table/smtp.chn* file. For example, to add two new hosts *chocolate* and *truffle*, enter:

```
chocolate.npr.COM: 123.456.789.4
truffle.npr.COM: 123.456.789.5
```

2. Now, edit the */usr/mmdf/table/smtp.dom* and match the unqualified SMTP host name to the fully qualified host name. For example, enter:

```
chocolate: chocolate.npr.COM
truffle: truffle.npr.COM
```

3. Now, use **dbmbuild**(ADM) to rebuild the MMDF hashed database.

Use the information in “Testing MMDF configuration” (page 769) to check the new SMTP configuration.

## Running multiple deliver daemons

If you configure MMDF to route mail on channels in addition to the local channel, you must restart the **deliver** daemon and modify the */etc/rc2.d/S86mmdf* system startup file to tell **deliver** about the new channels.<sup>8</sup> The **deliver** daemon periodically checks each channel’s queue for mail to deliver. This is known as “sweeping the queues”.

“Modifying MMDF system startup” (page 737) explains how to start one **deliver** process for all the channels that you have configured. When you have a single **deliver** program managing a number of channels, **deliver** goes through the channels individually and tries to deliver all the messages in a channel’s queue before going on to the next channel. You can configure your system to start multiple **deliver** programs, each servicing a single channel, instead. In this case, the **deliver** daemons work in parallel. This is a desirable configuration for a mail gateway machine because it increases the overall mail bandwidth of the machine.

**NOTE** For each **deliver** process that you initiate, system overhead increases.

\* See “Modifying MMDF system startup” (page 737) for information on how to do this.

To start separate **deliver** processes to manage each channel, use the following information to add **deliver** startup lines for each channel to the */etc/rc2.d/S86mmdf* file.

If you removed the **-clocal** from the **deliver** startup line in your *S86mmdf* file, the **deliver** startup line looks like the following:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b"
```

To create separate **deliver** processes, use the **-c** option to add lines for each channel. The following example shows separate **deliver** processes for each of the local, UUCP, and SMTP channels:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -clocal"
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -cuucp"
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -csmtp"
```

If you have a lot of mail traffic on a single channel, you can start more than one **deliver** process running on that channel. To do this, simply add more than one **deliver** startup line for the channel in */etc/rc2.d/S86mmdf*. For example, to start three **deliver** processes to sweep the queue for the SMTP channel, add these lines:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -csmtp"
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -csmtp"
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -csmtp"
```

Keep in mind that each **deliver** process that you start increases the load on the system.

### *Specifying different time intervals for deliver*

The **MSLEEP** parameter in the *mmdftailor* file determines the default time interval for the **deliver** daemon to sweep through the queues for all the channels. The default setting for **MSLEEP** is 600, meaning that **deliver** sweeps the queues every 600 seconds, or 10 minutes. We recommend that you set up **deliver** to run every 60 seconds. To use this interval for all the channels, change the value of **MSLEEP** in *mmdftailor* to 60.

If you create a separate **deliver** daemon for each channel, you can set up a different time interval for each **deliver** process. To do this, add the **-T** option to the appropriate **deliver** startup line in */etc/rc2.d/S86mmdf*. For example, to run **deliver** on the local channel every 30 seconds and on the UUCP channel every 5 minutes, put the following lines in *S86mmdf*:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -clocal -T30"
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -cuucp -T300"
```

## Rebuilding the MMDF hashed database

---

The hashed database gives MMDF quick access to alias and routing information. You must rebuild this database whenever you change *mmdftailor* or any of the alias, channel or domain tables.

To rebuild the database, log in as *mmdf* and run the following commands:

```
cd /usr/mmdf/table
./dbmbuild
```

If any of the files in */usr/mmdf/table* are missing when you rebuild the database, *dbmbuild* reports that they are missing. See the *dbmbuild(ADM)* manual page for full details.

## Testing MMDF configuration

---

Once you've modified your MMDF configuration and rebuilt the hashed database, use the information in this section to test the new configuration.

### Checking for MMDF problems

---

Use the *checkup(ADM)* command to examine the full MMDF system on the local machine and report any inconsistencies. Normally, *checkup* reports all problems, including correct states; *checkup* marks problems with two asterisk characters (\*\*) and encloses advisory information in square brackets. You can tell *checkup* to report only the problems using the *-p* option. For example, enter the following command:

```
/usr/mmdf/bin/checkup | more
```

The following example shows the output from *checkup*:

```
** Asterisks indicate potentially serious anomalies.
```

```
Tailor file : /usr/mmdf/mmdftailor
** Unknown tailor : 'MLCKTYPE advisor'
```

In this case, *checkup* does not recognize the *MLCKTYPE* parameter *advisor* because it is misspelled (the correct parameter name is *advisory*).

If the permissions on any of the MMDF configuration files are incorrect, *checkup* lists this information.

## Testing mail addresses

---

Use the `checkaddr(ADM)` command to test an individual address for validity. For example, to test the validity of the address, *andrei@scribe*, enter the following command:

```
/usr/mmdf/bin/checkaddr andrei@scribe
```

The `checkaddr` checks the address and displays the following message:

```
andrei@scribe: OK
```

This command is particularly useful for testing aliases after you have installed a new alias file. For more information, see the `checkaddr(ADM)` manual page.

## Maintaining the MMDF system

---

Once you configure MMDF, the MMDF system requires minimal system administration. In addition to maintaining the alias files (as described in “Maintaining user aliases” (page 760)) and adding new machines to your network (see “Modifying MMDF configuration” (page 759)) you must perform the following tasks:

- Monitor space in the */usr/spool/mail* directory where mail is received, and in the */usr/spool/mmdf* directory where queued mail is stored.
- Clean outdated mail from the mail queues.
- Remove log files periodically from the */usr/mmdf/log* directory.

The following sections describe the utilities you can use to maintain your MMDF system.

## Checking the status of mail queues

---

Use the `checkque(ADM)` program to check the status of the mail queues; this utility reports on the number of messages waiting for delivery. For example, enter the following command:

```
/usr/mmdf/bin/checkque
```

Here is an example of output from `checkque`:

```
Thu May 9 16:07: 2 queued msgs / 80 byte queue directory
 4 Kbytes in msg dir

 2 msgs 4 Kb (local) local : Local Delivery
 deliver start : Thu May 9 15:15
 deliver message : Thu May 9 15:15
 deliver end : Thu May 9 15:15 / 0 hours
*** WAITING ** First message : Wed Apr 24 16:03
```

If a queue is backed up with waiting mail, you can manually force **deliver**(ADM) to deliver the mail using a command like the following:

```
deliver -w -clist,uucp
```

The **-c** option specifies the channels to be processed and **-w** forces **deliver** and the channel programs to output informative messages as they try to deliver the mail. You can review the output for abnormalities, such as a rejected sender or recipient. For a complete description of **checkque**, see the **checkque**(ADM) manual page.

## Removing old mail from the queues

---

Use the **cleanque**(ADM) program to remove outdated files from the mail queues. Use **crontab**(C) as *mmdf* to run **cleanque** daily. You might want to run **cleanque** more often, depending on your mail volume. You can also run **cleanque** using **cron**(C). To do this, create a *crontabs* file for *mmdf* in the */usr/spool/cron/crontabs* directory that includes the following line:

```
0 0 * * 0-6 /usr/mmdf/bin/cleanque
```

You can also run **cleanque** by hand whenever you suspect a problem with mail delivery. The **cleanque** manual page provides a complete description of this program.

## Monitoring the size of log files

---

You should periodically check the size of log files in */usr/mmdf/log*. To limit the amount of log data that accumulates in a log file, move it to another filename, for example *chan.log-*.

If you move aside a log file, make sure that *mmdf* owns the new empty file. To back up the log file and create a new empty file, enter the following commands as *mmdf*:

```
cd /usr/mmdf/log
cp chan.log chan.log-
cat /dev/null > chan.log
```

## Advanced MMDF configuration tasks

---

This section describes some additional configuration tasks that you can use to customize your MMDF system. In general, most sites do not need to perform any of the configuration procedures described in this section.

## Changing the location of system mailboxes

---

By default, MMDf delivers mail to a file named with the user's login in the */usr/spool/mail* directory. If you want MMDf to deliver mail to a file or directory other than the default file, you can add lines (anywhere in the *mmdftailor* file) using the following **MDLVRDIR**, **MMBXNAME**, and **MMBXPROT** keywords.

Only one of **MDLVRDIR** and **MMBXNAME** can be non-null. If **MDLVRDIR** is null, MMDf delivers to the user's home directory; if **MMBXNAME** is null, MMDf uses the user's login as the name of the mailbox file. You can also change the location of the spool directory by setting **MDLVRDIR** to be non-null.

**MMBXPROT** sets the protection mode on mailbox files using the same set of octal numbers that you use to change access permissions on files with **chmod(C)**. For example:

```
MDLVRDIR ""
MMBXNAME ".mailbox"
MMBXPROT 0600
```

In this example, MMDf delivers to a *.mailbox* file in the user's home directory and sets the file protection so that only the owner can read or write to the file.

## Specifying the MMDf "signature"

---

You can change the signature that MMDf uses when notifying senders of mail delivery problems by setting the **MSIG** keyword in the */usr/mmdf/mmdftailor* file. This message should indicate which mail routing system was responsible. For example:

```
MSIG "MMDf Mail System"
```

With this setting, when users receive "Failed" mail messages from MMDf, the message appears to be from "MMDf Mail System".

## Configuring MMDf authorization

---

The MMDf authorization feature allows you to control the flow of mail through your host. For example, if your site is on the Internet (you registered your domain name with SRI), you might want to allow your employees (but no one else) to send mail from their home machines through your system to the Internet. In this case, you would want to set up MMDf to authorize mail from those users to pass through your machine, but prevent mail from other users from passing through.



With MMDF, you can control mail transferred through your host on a per-channel or per-user basis (or both). In other words, you can control the flow of mail on specific channels and from specific users.

To control authorization on a particular host, use the **auth** parameter to set an authorization level on the MCHN line in the `/usr/mmdf/mmdftailor` file. MMDF provides the seven levels of authorization listed in Table 14-8.

**Table 14-8** MMDF authorization levels

| Level    | Description                                                                                     |
|----------|-------------------------------------------------------------------------------------------------|
| free     | performs no authorization checks (default)                                                      |
| inlog    | logs incoming authorized and unauthorized access, but allows mail to pass                       |
| outlog   | logs outgoing authorized and unauthorized access, but allows mail to pass                       |
| inwarn   | logs incoming unauthorized access, transfers the message, and sends a warning to the originator |
| outwarn  | logs outgoing unauthorized access, transfers the message, and sends a warning to the originator |
| inblock  | logs the incoming unauthorized access attempt and bounces the mail                              |
| outblock | logs the outgoing unauthorized access attempt and bounces the mail                              |

The default channel authorization level is **free**; any channel that you do not set authorization for is set to **free**.

The following sections explain more about using these authorization levels to set up host- and user-based authorization.

### *Setting authorization for hosts*

You can control the flow of mail both to and from your host and network by setting up host-based authorization controls. The most common reason that sites use host-based authorization is to make sure that hosts which are not part of a private network do not use that network to send mail to other parts of the world. For example, you can use host-based authorization to allow mail to enter your domain, but restrict the mail passing through it.

By default, MMDF performs all authorization on the next “hop” host in the address; if someone specifies a route through your host, MMDF on your host can authorize the next hop in that route. In other words, MMDF does not perform authorization based on the final destination unless the final destination is the next hop. However, you can perform authorization on the entire route

to the destination or from the source using the **dho** flag with the **auth** parameter to specify routing-based authorization controls. See "Routing-based authorization" (page 775) for more information.

You can control access to a particular channel based on whether the mail is inbound or outbound on the channel and whether the source or the destination of the mail is authorized. To do this, use one (or more) of the four tables listed in Table 14-9.

**Table 14-9 Host-based authorization tables**

| Table          | Description                                                                                                                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>insrc</b>   | When mail arrives from a particular host (the source host), this table verifies that that host is allowed to send mail to the destination host by authorizing either the destination host or the channels used to access the destination host. |
| <b>outsrc</b>  | When mail arrives from the source host via a particular channel, this table determines if that source host or channel is allowed to send mail to the destination host.                                                                         |
| <b>indest</b>  | When mail is sent to a particular host or via a particular channel, this table determines whether the source host is allowed to send mail to the destination host or via that channel.                                                         |
| <b>outdest</b> | When mail is sent to a particular host, this table verifies that the source host or channel is allowed to send mail to the destination host.                                                                                                   |

For example, you might want to set up MMDF so that only hosts that belong to a certain network can send mail out on a specific channel. This could be the case if your site has limited resources and you do not want to allow other sites to pass mail through your site.

Use the following procedure to set up the host-based authorization controls:

1. First, log in as *mmdf*. (You must perform all MMDF configuration as *mmdf*.)
2. Use the **MTBL** keyword to define a table in */usr/mmdf/mmdftailor*. For example:

**MTBL "world-auth", file="authinfo/world"**

The *authinfo/world* table specifies the privileges for all the hosts that do not belong to the network. This table is stored in the file *world* in the directory */usr/mmdf/table/authinfo*; you can specify locations like this for all the tables.

- Now, use **MCHN** to define two channels in *mmdftailor*: one channel for the hosts that belong to the network, one for the hosts that do not belong.

For example, define a channel for the *sconet* network:

```
MCHN sconet, auth=free, show="SCONET Delivery",
 ap=822, mod=imm
```

(The "auth=free" authorization setting is the default; you do not have to specify it explicitly as in this example.)

Now, define a channel for all the hosts (**world**) not in the *sconet* network (this appears as one line in *mmdftailor*):

```
MCHN world, auth=inblock, indest="world-auth",
 show="WORLD Delivery", ap=822, mod=imm
```

In this case, anyone can send mail out on the **world** channel, but MMDF checks the *world-auth* table to authorize the destination of mail arriving on this channel.

- Now, create the channel tables (*\*.chn* files in the */usr/mmdf/table* directory) for the channels that you define in *mmdftailor* and include descriptions of each host accessed via that channel. (See "Channel files" (page 753) for more information.)
- Finally, create the *world-auth* table and include lines like the following:

```
world:
local:
sconet: moocow.uucp
```

The entries on the LHS of the authorization table specify that if the destination for a message is on either the **world** or **local** channels, MMDF authorizes anyone using the **world** channel as an input channel to send mail.

The entry on the RHS of the *sconet* channel entry specifies the hosts and channels that are authorized to send outgoing mail using the *sconet* channel. In this case, *moocow.uucp* is the only machine allowed to pass mail into the *sconet* network.

- Now, rebuild the hashed database using the information in "Rebuilding the MMDF hashed database" (page 738).

## ***Routing-based authorization***

You might want to set up authorization to keep mail that enters your domain from leaving your domain. In other words, people can send mail from inside the domain to people outside the domain as well as to other people in the domain, but people outside the domain cannot send mail through your domain to another destination.

For example, if your domain includes machines in different cities with links to different outside computers, people from outside the domain might use these links to send mail from one city to another, through your domain. In this case, you can set up MMDF authorization to prevent people from using your system to transfer mail.

Because MMDF performs authorization only on the next “hop” in the address, you cannot use the procedure described in the previous section to set up authorization control. To do this, you must use the `auth` parameter with the `dho` flag to specify routing-based authorization controls.

The following procedure explains how to set up routing-based authorization for the hosts that are not in the *nprnet* domain:

1. Log in as *mmdf*.
2. Define a table in */usr/mmdf/mmdftailor*. For example:  
**MTBL "world-auth", file="authinfo/world"**  
 This defines the *authinfo/world* file that contains the authorization information for the *world* channel.
3. Now, specify a channel for your domain. For example, if the domain name is *npr.com*, create an MCHN entry like the following:  
**MCHN nprnet, auth=free, show="NPRNET Delivery",  
 ap=822, mod=imm**
4. Now, define a channel for the rest of the hosts that are not in the local domain (again, this appears as one line in *mmdftailor*):  
**MCHN world, auth=inblock, auth=dho, indest="world-auth",  
 show="WORLD Delivery", ap=822, mod=imm**  
 The “*auth=indest*” parameter specifies that when *world* is the input channel, MMDF checks the *authinfo/world* file to verify that the inbound host is authorized to send mail to the destination.  
 When you specify the “*auth=dho*” parameter on a channel, MMDF replaces the “*host*” (in host-based authorization) used to check authorization with a route. The route is either from the source or to the destination, depending on which flag (from Table 14-9) that you specify. MMDF replaces the local section of the route (the user’s name) with the string “*username*”. Then, MMDF compares this route to the route specified in the message, to determine if the message is authorized or not.
5. Create the channel tables in the */usr/mmdf/table* directory for the channels that you define in *mmdftailor*. (See “Channel files” (page 753) for more information.)

6. Finally, create the *authinfo/world*, and include entries like the following:

```
world:
username@npr.com:
username@larry.npr.com:
username@moe.npr.com:
username@curly.npr.com:
```

This table authorizes MMDF to deliver any mail addressed to people in the *npr.com* domain arriving or leaving on the **world** channel. This does not allow mail to pass through the **nprnet** channel to a destination outside the *npr.com* domain.

7. Now, rebuild the hashed database with **dbmbuild**.

### Setting authorization for users

To configure MMDF authorization on a per-user basis, you must first set up the authorization level on the channels that you want to restrict as described in the previous section. Once you set up channel authorization levels, you must create a table in the */usr/mmdf/table* directory that maps user names to authorization levels and then declare this table in the */usr/mmdf/mmdftailor* file.

The following is the format for setting up the per-user authorization table:

*username: keyword channel [, channel]*

The *username* can be either a local or remote user name, *keyword* is one of the keywords for the actions that you can authorize users to perform, and *channel* is the channel name from the MCHN line.

Table 14-10 lists the action authorization keywords:

**Table 14-10** MMDF authorization keywords

| Keyword | Description                                                                              |
|---------|------------------------------------------------------------------------------------------|
| both    | allows user to send and receive mail                                                     |
| send    | allows user to send mail only                                                            |
| recv    | allows user to receive mail only                                                         |
| expire  | expires access privileges for the user (and includes this information in any error mail) |

MMDF treats any action not listed in the table above as **expire**, except that MMDF sends the text of the action instead of "expire" to the user in error mail.

Use these steps to set up authorizations for specific users:

1. First, log in as *mmdf*.
2. Use the information in "Setting authorization for hosts" (page 773) to set up authorization for any channels that you want to restrict.

3. Declare the user authorization table in the `/usr/mmdf/mmdftailor` file. For example, if the name of the user authorization table is `auth.user`, the MTBL declaration looks like this:

```
MTBL auth, file="auth.user", show "Per-user authorization"
```

Note that you must call the per-user authorization table "auth"; MMDF treats any table called "auth" as the per-user authorization table.

4. Now, create the user authorization table in the `/usr/mmdf/table` directory. Edit the file and include one line for each user to whom you want to grant mail access. Any users not listed in the user authorization table are not authorized to use any channel except the ones set to the **free**, **inlog**, or **out-log** authorization levels.

In our example, MMDF allows users who are not listed in the user authorization table to pass mail in and out through the UUCP channel and in through the TCP/IP channel.

5. To set up access authorizations for a local user, specify the unqualified user name. In the following example, the local user *andrei* can both send and receive mail on the TCP/IP and UUCP channels:

```
andrei: both smtp,uucp
```

However, if you set up host-based authorization to restrict access to one of these channels, for example UUCP, *andrei* might not be authorized to send or receive mail on that channel.

Because mail on the local channel is not restricted, *andrei* can pass mail through this channel even though the user authorization list does not include "local" in the list of channels.

6. To set up access authorization for a remote user, specify the fully qualified address of that user. For example, to allow *natasha* on the machine *kgb.GOV* to send mail through TCP/IP on this host, add a line like the following to the authorization table:

```
natasha@kgb.GOV: send smtp
```

If mail arrives for *natasha* through UUCP, or if she tries to send mail through the UUCP channel, MMDF rejects the mail.

If you want to expire a particular user's access and tell MMDF to send an error message, add the message to the user authorization table line for that user. For example, to expire *rocky@squirrel.COM*'s access and send the text "No more mail for you!", include a line like this one:

```
rocky@squirrel.COM: "No more mail for you!" uucp
```

7. Rebuild the hashed database with **dbmbuild**.

## Setting host and user authorization controls

The default authorization algorithm allows MMDF to deliver a message if either the user-based or host-based authorization tables allow it. Thus, if a user table authorizes a user to send mail over a channel, but the host table does not, that user can still use the channel (and vice versa).

If you want to require that both the user and host tables authorize access to a channel, use the `hau` keyword with the `auth` parameter in the `MCHN` channel declaration in `mmdftailor`.

## Understanding authorization log files

If you set the `level` parameter to the `AUTHLOG` keyword to an authorization level of `FST` (to give full statistics) or higher<sup>9</sup>, MMDF maintains a complete log of authorization attempts and reasons for failure or success.

For example, to set the authorization logging level to save all diagnostic messages to the `/usr/mmdf/log/mmdfauth.log` file, set `AUTHLOG` in the `mmdftailor` file like this:

```
AUTHLOG /usr/mmdf/log/mmdfauth.log level=FTR
```

The format of authorization messages in the log file is similar to other MMDF log files. Each message includes the date, time, message source, and message ID, followed by the log-specific information. For example:<sup>10</sup>

```
4/29 9:44:54 AU-0000: msg.a000561: i='local' o='ucsc'
a='lisa@rsre.AC.UK' r='CH' hi='' ho='username@rsre.ac.uk'
4/29 9:44:54 AU-0000: msg.a000561: i='local' o='ucsc'
a='jane@rsre.AC.UK' r='CH' hi='' ho='username@rsre.ac.uk'
4/29 9:44:55 AU-0000: msg.a000561: END size='2102', sender='robert'
```

Table 14-11 describes the authorization message keys.

**Table 14-11 Authorization keys**

| Key | Description              |
|-----|--------------------------|
| i   | input channel            |
| o   | output channel           |
| a   | destination address      |
| r   | reason for authorization |
| hi  | inbound host             |
| ho  | outbound host            |

\* See the section on `MCHANLOG` in the `mmdftailor(F)` manual page for a complete list of logging levels.

† The authorization messages appear on one line; the examples in this section split the lines for readability.

In addition, the authorization message includes an “end of processing” message that describes the message sender and size. (In the example above, the third line is the “end of processing” message.)

Each authorization message can have either one or two reasons for authorizing a particular message. If the authorization line includes one reason, the “r” key specifies a single authorization code which describes both the inbound and the outbound authorization when you use host-based authorization. These codes are listed Table 14-12.

**Table 14-12 Single-reason authorization codes**

| Code | Description                                |
|------|--------------------------------------------|
| OH   | outbound host/route                        |
| HC   | outbound host/route and inbound channel    |
| HH   | inbound host/route and outbound host/route |
| CC   | inbound channel and outbound channel       |
| CH   | inbound channel and outbound host/route    |
| IH   | inbound host/route                         |

The following example uses the CH authorization code:

```
4/29 9:44:54 AU-0000: msg.a000561: i='local' o='peaks'
a='bob@rsre.AC.UK' r='CH' hi='' ho='username@rsre.ac.uk'
4/29 9:44:54 AU-0000: msg.a000561: i='local' o='peaks'
a='mike@rsre.AC.UK' r='CH' hi='' ho='username@rsre.ac.uk'
4/29 9:44:55 AU-0000: msg.a000561: END size='2102', sender='cooper'
```

In this example, the authorized message has two recipients (*bob* and *mike*). The first authorization message shows that the inbound channel (“i”) is the local channel and the outbound channel is **peaks**. The “a” key indicates that the recipient’s address is *bob@rsre.AC.UK*.

The reason (“r”) given for authorizing the message is CH; in other words, the inbound channel (**local**) has authorization to send mail to the given outbound host or route (specified by “ho”), in this case *username@rsre.ac.uk*.

The two-reason authorization codes describe the reason for authorization in terms of user-based authorization. Table 14-13 lists these codes.



**Table 14-13 Two-reason authorization codes**

| Code | Description                                |
|------|--------------------------------------------|
| IL   | inbound channel, outbound = LIST           |
| OL   | outbound channel, dest = LIST              |
| IS   | inbound channel by sender                  |
| OS   | outbound channel by sender                 |
| IR   | inbound channel by receiver                |
| OR   | outbound channel by receiver               |
| I    | inbound channel, log unauthorized access*  |
| O    | outbound channel, log unauthorized access* |

\* MMDF only uses these authorization codes when you set "auth=inlog" or "auth=outlog".

The authorization message in the following example uses two-reason authorization (if no authorization is required for a channel, MMDF leaves the reason field ("r") empty):

```
4/29 9:53:09 AU-0000: msg.a000653: i='local' o='npr'
a='john@edxa.ac uk' r='' r='OS'
4/29 9:53:10 AU-0000: msg.a000653: END size='197', sender='david'
```

In this example, the message arrived (with no authorization required) on the local channel and is authorized to leave on the **npr** channel because the sender (*david*) is authorized to use it (OS).

## Adding a new alias file

It is much easier to add entries to an existing alias file than to create one; see "Alias files" (page 747) and "Maintaining user aliases" (page 760) for more information. If you want to add a new alias file, use the following procedure:

1. Log in as *mmdf*.
2. Create a file in the */usr/mmdf/table* directory.
3. Verify that the new alias file is owned and group-owned by *mmdf* and set the permissions to 644.

**NOTE** Do not give write permissions on alias files to other users. With write permissions to alias files, a user can forward another user's mail to another location. For this reason, you should designate a trusted user to make changes to the alias files as requested.

4. Add alias information to the new alias file. See "Alias files" (page 747) for information on the format of the various alias files.

- Now, define the alias file in the */usr/mmdf/mmdftailor* file. Create an MTBL line to associate an abbreviated name and a more descriptive name with the new alias file. For example, the MTBL line in *mmdftailor* defines the alias file *alias.user* as:

```
MTBL auser, file="alias.user", show="User Aliases"
```

Now, you can refer to the file */usr/mmdf/table/alias.user* as "auser" throughout the rest of *mmdftailor*.

- Add an ALIAS entry to *mmdftailor* to define information about the new alias file. For example, here is the ALIAS entry for *alias.user*:

```
ALIAS table=auser
```

The default parameters in the *mmdftailor* file are adequate for most requirements; however you can specify characteristics for the alias file using the following parameters:

**nobypass** specifies that MMDF search the alias files first for all mail addresses on outgoing or incoming mail. In MMDF, placing a tilde (~) before an address causes the address to be interpreted as destined for its literal meaning. The **nobypass** parameter means that normal aliasing cannot be bypassed. For example, suppose the alias file contains this entry for George and **nobypass** is not set:

```
george: george@vegan.edu
```

If you enter:

```
mail george
```

then the mail goes to the *george* login on the *astoria* machine. If you then enter:

```
mail "~george"
```

then aliasing is disabled and mail is sent to a George on the local machine. If **nobypass** is set, entering the tilde character (~) has no effect and every mail request is translated by the alias files.

**public** allows remote sites to expand aliases from this table using the SMTP EXPN command; you can also use **malias(C)** to display this information.

**trusted** directs mail to be delivered to any file or process using the permissions of any user on the system (including *root*); only the super user should have access to modify a trusted alias file.

Note that MMDF searches the alias tables in the order that you list them in *mmdftailor*.

For more information, see the **mmdftailor(F)** manual page.

## Adding a new domain file

---

Use the following procedure to create a new domain file:

1. Log in as *mmdf*.
2. Create a file in the */usr/mmdf/table* directory. Generally, you should name domain files for the domain that they describe. For example, the domain file for the *npr.COM* domain is called *npr.dom*.
3. Verify that the new domain file has the correct permissions (see step 3 in the previous section).
4. Add the information for the domain to the new domain file. See "Domain file format" (page 751) for more information.

**NOTE** The entries on the left-hand side in this file must contain no leading whitespace.

5. Create an MTBL entry in *mmdftailor* for the new domain file. For example:

```
MTBL nprdom, file='npr.dom', show='NPR Domain'
```

See the *mmdftailor(F)* manual page for more information about additional parameters that you can use to specify additional properties about the table that you are defining with MTBL.

6. Add an MDMN entry for the new file:

```
MDMN 'npr.COM', show='NPR Domain', table=nprdom
```

The first argument is the name of the domain.

## Adding a third-party channel program

---

If you add a new channel program from an SCO add-on package, you must define and add a new channel file. A channel is the mechanism for delivering mail either to a mailbox on the local machine or across the network to a remote machine. The MCHN entries in *mmdftailor* define the channels available to MMDF for mail transport.

You can define additional channels for the network protocols configured on your system by specifying them in the *mmdftailor* file. Channel definitions look like this:

```
MCHN uucp, show='UUCP Delivery', que=uucp,
 tbl=uuchn, ap=822, pgm=uucp, mod=imm
```

The first argument on the MCHN line is the name of the channel.

Use these steps to add a new channel:

1. Create an MCHN entry in */usr/mmdf/mmdftailor* and specify the parameters to use in your channel definition. For a complete list of the MCHN parameters, see the *mmdftailor(F)* manual page.
2. Create a channel file (*\*.chn*) in */usr/mmdf/table* and include entries for each host that you access via the new channel and how to route mail to those hosts. For more information, see "Channel files" (page 753).

## Changing MMDF parameters

---

The MMDF parameters in the */usr/mmdf/mmdftailor* file allow you to redefine the certain MMDF variables. This section covers some of the more common MMDF parameters; for a complete list, see the *mmdftailor(F)* manual page.

**AUTHLOG** controls the authorization information. The following example shows the format of this parameter:

```
AUTHLOG /tmp/mmdf/mmdfauth.log, level=FST, size=40, stat=some
```

The **AUTHLOG** level must be at least FST, or MMDF does not save any authorization logging information. For more information, see the section on **MCHANLOG** in the *mmdftailor(F)* manual page.

**MFAILTIME** is the time (in hours) a message can remain in a queue before MMDF sends a failed mail message to the sender and purges the message from the queue.

```
MFAILTIME 168
```

**MLCKTYPE** allows you to specify the locking protocol for MMDF to use when locking users' mailboxes. This is useful if the users on the system use third-party MUA's that use a lock file that is different from the standard System V lock file. Set the **MLCKTYPE** parameter to one or more of the keywords in Table 14-14.

**Table 14-14 MMDF locking keywords**

| Keyword  | Lock file                                                       |
|----------|-----------------------------------------------------------------|
| advisory | System V <code>fcntl()</code> kernel locking protocols          |
| v7       | Version 7 and System V Release 3, and earlier locking protocols |
| xenix    | XENIX locking protocols                                         |
| all      | all above locking protocols (default)                           |

The default locking protocol is **all**; however the MMDF configuration utility sets **MLCKTYPE** to **advisory**.

With the **v7** keyword, MMDF creates a file called *username.lock* in */usr/spool/mail*. The **xenix** keyword specifies that MMDF create a file called */tmp/username.mlk*. In both cases, *username* is the name of the user's mailbox.

You can specify more than one locking protocol on the **MLCKTYPE** line. If you do, MMDF must successfully lock the user's mailbox using all the locking protocols before the mailbox is considered locked. For example, to use **advisory** and **xenix**, set **MLCKTYPE** like the following:

```
MLCKTYPE advisory, xenix
```

In this case, MMDF must lock the mailbox, using the **fcntl()** kernel file locking protocol *and* create a file called */usr/spool/mail/username.lock*. If it fails to perform both of these locks, MMDF must release the successful lock and try again later. Thus, if MMDF failed to perform the **fcntl** lock, it must unlink */usr/spool/mail/username.lock*.

**MMAXHOPS** specifies the maximum number of "Received:" or "Via:" lines a message can contain before the MMDF considers that the message is looping and rejects it.

```
MMAXHOPS 20
```

**MWARNTIME** specifies the time (in hours) that a message can remain in a queue before MMDF sends a warning message about delayed delivery to the sender.

```
MWARNTIME 72
```

**MMSGLOG** controls the logging information produced by the **deliver**(ADM) and **submit**(ADM) programs. The following example shows the format of this parameter:

```
MMSGLOG /tmp/mmdf/mmdfmsg.log, level=FST, size=40, stat=some
```

For more information, see the **logs(F)** manual page and the section on **MCHANLOG** in the **mmdftailor(F)** manual page.

**MCHANLOG** controls MMDF logging, except for information controlled by **AUTHLOG** and **MMSGLOG**. See the **mmdftailor(F)** manual page for details.

**MSLEEP** determines the length of time (in seconds) that the **deliver** daemon sleeps between scanning the queues. By default, **MSLEEP** is set to 600 (10 minutes). We recommend that you set **MSLEEP** to 60 seconds:

```
MSLEEP 60
```

## Getting more information

If you need more information to understand how the various components of MMDF interact, refer to the manual pages listed in Table 14-15.

**Table 14-15** MMDF manual pages

| Manual page            | Description                                                 |
|------------------------|-------------------------------------------------------------|
| <b>checkaddr(ADM)</b>  | checks a mail address for validity                          |
| <b>checkqueue(ADM)</b> | reports on MMDF queue status                                |
| <b>checkup(ADM)</b>    | checks MMDF system configuration                            |
| <b>cleanqueue(ADM)</b> | sends warnings and returns expired mail                     |
| <b>cnvtnbox(ADM)</b>   | converts XENIX-style mailboxes to MMDF format               |
| <b>dbmbuild(ADM)</b>   | builds the MMDF database                                    |
| <b>dbmedit(ADM)</b>    | edits the MMDF database                                     |
| <b>deliver(ADM)</b>    | delivers mail to MMDF channels                              |
| <b>list(ADM)</b>       | handles mailing lists                                       |
| <b>mmddf(ADM)</b>      | routes mail locally or over network                         |
| <b>mmdfalias(ADM)</b>  | converts XENIX-style aliases file to MMDF                   |
| <b>rmail(ADM)</b>      | submits remote mail received via MMDF                       |
| <b>submit(ADM)</b>     | accepts mail to be handled by MMDF                          |
| <b>uucico(ADM)</b>     | transfers UUCP files                                        |
| <b>uulist(ADM)</b>     | converts UUCP file to MMDF format                           |
| <b>checkmail(C)</b>    | checks for submitted but not delivered mail                 |
| <b>mail(C)</b>         | mails messages between users                                |
| <b>uucp(C)</b>         | sends information between UNIX machines over serial lines   |
| <b>rcvalert(C)</b>     | alerts user when mail is received                           |
| <b>rcvfile(C)</b>      | stores mail in a file automatically                         |
| <b>rcvprint(C)</b>     | prints received mail automatically                          |
| <b>rcvtrip(C)</b>      | informs sender that user is away                            |
| <b>uname(C)</b>        | prints the name of the current system                       |
| <b>uux(C)</b>          | transfers UUCP files and executes commands on remote system |
| <b>logs(F)</b>         | logs MMDF messages                                          |
| <b>maildelivery(F)</b> | describes the user delivery specification file              |
| <b>mmdftailor(F)</b>   | provides run-time tailoring                                 |
| <b>queue(F)</b>        | queues files for storing mail in transit                    |
| <b>systems(F)</b>      | UUCP Systems file format                                    |
| <b>tables(F)</b>       | describes alias, domain, and host tables                    |
| <b>llog(S)</b>         | performs standardized information logging for programs      |

*(Continued on next page)*

**Table 14-15** MMDF manual pages  
(Continued)

| Manual page       | Description                                    |
|-------------------|------------------------------------------------|
| <b>ml_send(S)</b> | provides simple interface for mail submission  |
| <b>mmdf(S)</b>    | simplifies MMDF MUA mail submission and pickup |
| <b>phs(S)</b>     | notes the MMDF transmission phase              |
| <b>tai(S)</b>     | gets site tailoring information                |

## Chapter 15

# *sendmail administration*

---

The **sendmail(ADMN)** utility provides a “back-end” mail interface to other mail programs; that is, **sendmail** only routes and delivers mail, it does not provide a user interface for composing or formatting letters. **sendmail** provides network access and supports either Internet-style addressing, such as **user@domain** or UUCP-style addressing, such as **host!user**. **sendmail** works with delivery agents such as SMTP (Simple Mail Transfer Protocol), X.400, and UUCP. **sendmail** only handles text format files; binary data is not allowed.

**NOTE** This chapter describes how you can manually fine-tune **sendmail** configuration. In most installations, running the initialization script **mkdev sendmail-init** without any manual configuration will produce the desired results on your system. In that case, the information found in this chapter following the section “Configuring a standard installation” is unnecessary, and is provided for completeness only.

You do not need to use **sendmail** to route mail; the operating system’s MMDF mail router is also available and provides several key benefits over using **sendmail**, including an easier configuration process. For more information on MMDF, refer to the previous chapter of this guide.

This chapter compares **sendmail** to other mailers, discusses how **sendmail** works, explains **sendmail** configuration, shows how to run **sendmail**, and details **sendmail** administration.



## *sendmail and other mailers*

---

This section compares **sendmail** with three other mail programs:

- **delivermail**
- MMDF (Multichannel Memorandum Distribution Facility)
- MPM (Message Processing Module)

These comparisons are provided for those who are familiar with these other mail routing programs.

### *Comparing sendmail with delivermail*

---

The **sendmail** program is an outgrowth of **delivermail**. The primary differences are as follows:

- Configuration information is not compiled in. This change simplifies many of the problems of moving to other machines. It also simplifies debugging of new mailers.
- Address parsing is more flexible. For example, **delivermail** only supported one gateway to any network, whereas **sendmail** can be sensitive to host names and reroute to different gateways.
- The forward and include features of **sendmail** eliminate the requirement that the system alias file be writable by any user (or that an update program be written, or that the system administration make all changes).
- The **sendmail** program supports message batching across networks when a message is being sent to multiple recipients.
- A mail queue is provided in **sendmail**. Mail that cannot be delivered immediately but can potentially be delivered later is stored in this queue for a later retry. The queue also provides a buffer against system crashes; after the message has been collected, it may be reliably redelivered even if the system crashes during the initial delivery.
- The **sendmail** program uses the networking support of the operating system to provide a direct interface to networks such as Ethernet using SMTP (the Simple Mail Transfer Protocol) over a TCP/IP connection.

### *Comparing sendmail with MMDF*

---

The Multichannel Memorandum Distribution Facility (MMDF) spans a wider problem set than **sendmail**. For example, the domain of MMDF includes a “phone network” mailer, whereas **sendmail** calls on pre-existing mailers in most cases.

MMDF and **sendmail** both support aliasing, customized mailers, message batching, automatic forwarding to gateways, queueing, and retransmission. MMDF supports two-stage timeout, which **sendmail** does not support. MMDF uses two-stage timeout when routing mail through machines to users. If a message cannot be forwarded to a particular machine or to a particular user on a machine, a warning is sent back to the mail message sender. This is stage 1. At some future time (configurable by the administrator), the message is relayed again. If it fails, a failure message is returned to the sender, and MMDF makes no further attempts to resend the original message. This is stage 2.

MMDF does offer several substantial benefits over **sendmail**, including:

- Configuration files that are easy to read and understand.
- The ability for end-users to configure their own sorting parameters.
- A larger set of supported delivery agents.

Because MMDF does not consider backwards compatibility as a design goal, the address parsing is simpler but much less flexible.

It is somewhat harder to integrate a new channel into MMDF. In particular, MMDF must know the location and format of host tables for all channels, and each channel must speak a special protocol. This allows MMDF to do additional verification (such as verifying host names) at submission time.

MMDF strictly separates the submission and delivery phases. Although **sendmail** has the concept of each of these stages, they are integrated into one program, whereas in MMDF they are split into two programs.

## *Sendmail and the message-processing module (MPM)*

---

The Message Processing Module (MPM) matches **sendmail** closely in terms of its basic architecture. However, like MMDF, the MPM includes the network interface software as part of its domain.

MPM also postulates a duplex channel to the receiver, as does MMDF. This allows simpler handling of errors by the mailer than is possible in **sendmail**. When a message queued by **sendmail** is sent, any errors must be returned to the sender by the mailer itself. Both MPM and MMDF mailers can return an immediate error response, and a single error processor can create an appropriate response.

MPM prefers passing the message as a structured object, with (type, length, value) triples. Such a convention requires a much higher degree of cooperation between mailers than is required by **sendmail**. MPM also assumes a universally agreed-upon Internet name space (with each address in the form of a net-host-user tuple), which **sendmail** does not.

## *How sendmail works*

---

When a sender, for example the `mail(C)` command, wants to send a message, a request is issued to `sendmail` directly, via SMTP over a pair of pipes, or via SMTP using a socket. `sendmail` then collects and queues the message along with other messages, and when ready delivers the message. If any errors occur while delivering the message, `sendmail` returns a message to the sender describing the error. Alternatively, it may return a status code to indicate what went wrong.

If `sendmail` is called using SMTP over pipes or through a socket, the arguments are first scanned and option specifications are processed. Recipient addresses are then collected, either from the command line or from the SMTP command `RCPT` (recipient), and a list of recipients is created. Aliases are expanded at this point. This includes any aliases that are part of a mailing list. At this stage, as much validation of the addresses as possible is done. Syntax is checked and local addresses are verified, but detailed checking of host names and addresses is deferred until delivery. Forwarding is also performed as the local addresses are verified.

The `sendmail` program appends each address to the recipient list after parsing the recipient list. When a name is aliased or forwarded, the old name is retained in the list, and a flag is set that tells the delivery phase to ignore this recipient. This list is kept free from duplicates, preventing alias loops and duplicate messages from being delivered to the same recipient, as might occur if a person is in two groups.

## *Collecting messages*

---

The `sendmail` program collects the message after the address parsing is complete. The message should have a header at the beginning. No formatting requirements are imposed on the message, except that they must be lines of text; binary data is not allowed. A message consists of a header and a body, which are separated by a blank line. The header is parsed and stored in memory, and the body of the message is saved in a temporary file.

To simplify the program interface, the message is collected even if no addresses are valid. The message is then returned to the sender with an error.

The header is formatted as a series of lines of the form:

*field-name: field-value*

*field-value* can be split across lines by starting the lines that follow with a space or a tab. Some header fields have special internal meaning, in which case they are subject to special processing. Other headers are simply passed through. Some header fields, such as time stamps, may be added automatically.

The message body is a series of text lines. It is completely uninterpreted and untouched by **sendmail**, except that lines beginning with a dot "." have the dot doubled when transmitted over an SMTP channel. This extra dot is stripped by the receiver. (SMTP uses lines beginning with a dot to signal the end of the message.)

## *Delivering messages*

---

For each unique mailer and host in the recipient list, **sendmail** calls the appropriate mailer. Each mailer invocation sends to all users receiving the message on one host. Mailers that accept only one recipient at a time are handled accordingly.

The message is sent to the mailer, using one of the same three interfaces used to submit a message to **sendmail**. Each copy of the message is prefaced by a customized header. The mailer status code is caught and checked, with a suitable error message given as appropriate. The exit code must conform to a system standard; otherwise, a generic message such as Service unavailable is given.

The send queue is sequenced by the receiving host before transmission in order to implement message batching. Each address is marked as it is sent, and so rescanning the list is safe. An argument list is built as the scan proceeds. Mail to files is detected during the scan of the send list.

After a connection is established, **sendmail** makes the changes to the header necessary for correct interpretation by a particular mailer and sends the result to that mailer. If any mail is rejected by the mailer, a flag is set to invoke the return-to-sender function after all delivery completes.

## *Queueing for retransmission*

---

If the mailer returns a status code indicating that the message should be sent later, **sendmail** puts the mail on the queue and tries again later.

## *Return to sender*

---

If errors occur during processing, **sendmail** returns the message to the sender for retransmission. If the user agent (mail) detects the error, then it is put in the *dead.letter* file located in the sender's home directory. If a **sendmail** server is connecting with a **sendmail** client on another machine, then the user is presumed to have become detached from the transaction, and so the message is mailed back to them.

## *Editing the message header*

---

A certain amount of editing occurs automatically to the message header. Header lines can be inserted under control of the configuration file. Some lines can be merged. For example, a "From:" line and a "Full-name:" line can be merged under certain circumstances.

## *Aliasing, forwarding and including mail*

---

The **sendmail** program reroutes mail in any of the following three ways:

- by aliasing
- by forwarding
- by inclusion

Aliasing applies across the entire system. Forwarding allows all users to reroute incoming mail destined for their accounts. Inclusion directs **sendmail** to read a file for a list of addresses. Forwarding is normally used in conjunction with aliasing. Each of these methods is described in more detail in the next three sections.

### *Aliasing*

Aliasing matches names to address lists using a system-wide file. This file is indexed to speed access. Only names that parse as local are allowed as aliases. This guarantees a unique key. The alias file is usually configured to be */usr/lib/aliases*. This file is not in the same format as the alias file */usr/lib/mail/aliases*. The identity of the alias file is configured through the *sendmail.cf* file.

### *Forwarding*

After aliasing, recipients that are local and valid are checked for the existence of a *forward* file in their home directory. If one exists, the message is not sent to that user, but rather to the list of users in that file. Often, this list contains a single address, and is used only for network mail forwarding.

Forwarding also permits a user to specify a private incoming mailer. For example, this forwarding uses a different incoming mailer:

```
" | /usr/local/newmail myname"
```

### *Including*

The syntax for including a file is:

```
:include: pathname
```

An address of this form reads the file specified by *pathname* and sends to all users listed in that file.

The intention is not to support direct use of this feature, but rather to use this as a subset of aliasing. In the following example, the form of the alias used is a method of letting a project maintain a mailing list without interaction with the system administration, even if the alias file is protected.

```
project: :include:/usr/project/userlist
```

It is not necessary to rebuild the index on the alias database when a list of this type is changed. All that is needed is to edit the include file to reflect the changes. In this example, the include file is */usr/project/userlist*.

## Queued messages

---

If the mailer returns a temporary failure exit status, the message is queued. A control file describes the recipients to be sent to and various other parameters. This control file is formatted as a series of lines, each describing a sender, a recipient, the time of submission, or some other significant parameter of the message. The header of the message is stored in the control file, so that the associated data file in the queue is just the temporary file that was originally collected.

## Configuring sendmail

---

There are three basic steps to installing **sendmail**. They are:

- installing the software, as described in the SCO system *Installation and Upgrade Guide*.
- running **mkdev sendmail-init** to create the **sendmail** configuration file
- editing the **sendmail** configuration file to fine-tune **sendmail** operation

Because the basic configuration file is adequate for most sites, the third step is often unnecessary.

The configuration file is read by **sendmail** reads when the program starts. This file describes the mailers that **sendmail** knows about, how to parse addresses, how to rewrite the message header, and the settings of various options.

## Configuring a standard installation

---

To use the **sendmail** mail system, you must first run **mkdev sendmail-init**, which initializes **sendmail** and configures it by invoking **mkdev cf**. After first initializing **sendmail**, you can reconfigure it at any time by running **mkdev cf**.

The **mkdev cf** script supports both UUCP and Internet style addressing. The UUCP configuration allows a network administrator to designate a machine on the network as the UUCP Gateway Machine, which will process all UUCP requests for the net or subnet.

To configure a standard installation with **mkdev sendmail-init**:

1. Type **mkdev sendmail-init** at the command line. The message Saving the following files: appears, followed by a list of files and this message: The mail system will now use sendmail for delivering messages.  
The **mkdev sendmail-init** command installs and configures **sendmail**. The **mkdev cf** command reconfigures **sendmail**.
2. A list of files is displayed, along with the message: Press <Return> to continue.
3. A menu appears. You must go through the various options of the menu in sequence.
4. The first menu option is to edit UUCP connections. Default values can be taken from the system, but you need to know if your machine is a UUCP gateway or not.
5. The second menu option is to edit the domain name. The default domain name is taken from the **hostname** utility; otherwise, you can specify a domain name.
6. The third menu option is to edit alternate host names. This menu selection is optional.
7. The fourth menu option is to edit network configuration information. The default answers are all "no" for the questions displayed in this option. Change the answers if necessary. If your machine is a UUCP mail gateway, then answer "yes" to the question Is there a UUCP gateway?.
8. The fifth menu option is simply a display that allows you to review your answers from the preceding options. If necessary, go back and correct any errors at this point by returning to the earlier option. This menu selection is optional.
9. The sixth menu option generates the new *sendmail.cf* file. A display of your selections is presented first, then you can generate the new file.  
You are prompted with this message: Do you wish to change anything? [y/n] The old *sendmail.cf* file is saved as */usr/lib/custom/save/sendmail.cf*.
10. The seventh menu option lets you quit the **sendmail** configuration menu.

## *Configuring a non-standard installation*

---

If the default configuration performed by **mkdev sendmail-init** is not adequate for your system, you will need to edit the **sendmail** configuration file, *sendmail.cf*, then run **mkdev cf** to effect the changes you made. This section describes the configuration file in detail, including hints on how to write one of your own if you have to.

There is one point that should be made clear immediately: the syntax of the configuration file is designed to be reasonably easy to parse, because this is done every time **sendmail** starts up. As a result, the configuration file is not particularly easy for a human to read or write.

An overview of the configuration file is given first, followed by details of the semantics.

## *The syntax*

The configuration file is organized as a series of lines, each of which begins with a single character defining the semantics for the rest of the line. Lines beginning with a space or a tab are continuation lines (although the semantics are not well defined in many places). Blank lines and lines beginning with a number sign “#” are comments.

### **R and S - rewriting rules**

The core of address parsing is the rewriting rules. These are an ordered production system. The **sendmail** command scans through the set of rewriting rules looking for a match on the left-hand side (*lhs*) of the rule. When a rule matches, the address is replaced by the right-hand side (*rhs*) of the rule.

There are several sets of rewriting rules. Some of the rewriting sets are used internally and must have specific semantics. Other rewriting sets do not have specifically assigned semantics, and may be referenced by the mailer definitions or by other rewriting sets.

The syntax of these two commands is:

**Sn**                Sets the current ruleset being collected to *n*. If you begin a ruleset more than once, it deletes the old definition.

### **Rlhs rhs comments**

The *lhs* is a pattern that is applied to the input. If it matches, the input is rewritten to the *rhs*. The *comments* are ignored. The fields must be separated by at least one tab character; there may be embedded spaces in the fields.

### **D - define macro**

Macros are named with a single character. These can be selected from the entire ASCII set, but user-defined macros should be selected from the set of uppercase letters only. Lowercase letters and special symbols are used internally.

The syntax for macro definitions is:

**Dx val**

Here *x* is the name of the macro and *val* is the value it should have. Macros can be interpolated in most places using the escape sequence **\$x**.



**C and F - define classes**

Classes of words may be defined to match on the left-hand side of rewriting rules. For example, a class of all local names for this site might be created so that attempts to send to oneself can be eliminated. These can either be defined directly in the configuration file or read in from another file. Classes may be given names from the set of uppercase letters. Lowercase letters and special characters are reserved for system use.

The syntax is:

**Cc** *word1 word2...*

**Fc** *file*

The first form defines the class *c* to match any of the named words. It is permissible to split them among multiple lines, for example:

CHmonet ucbmonet

This example is equivalent to the following:

CHmonet

CHucbmonet

The second form reads the elements of the class *c* from the named *file*.

**M - define mailer**

Programs and interfaces to mailers are defined in this line. The format is:

**Mname**, {*field=value*}\*

Here *name* is the name of the mailer (used internally only) and the "*field=name*" pairs define attributes of the mailer. Fields are:

|           |                                            |
|-----------|--------------------------------------------|
| Path      | pathname of the mailer                     |
| Flags     | special flags for this mailer              |
| Sender    | rewriting set for sender addresses         |
| Recipient | rewriting set for recipient addresses      |
| Argv      | argument vector to pass to this mailer     |
| Eol       | end-of-line string for this mailer         |
| Maxsize   | The maximum message length for this mailer |

Only the first character of the field name is checked.

**H - define header**

The format of the header lines that *sendmail* inserts into the message is defined by the H line. The syntax of this line is:

**H[?mflags?]***hname: htemplate*

Continuation lines in this specification are reflected directly into the outgoing message. The *htemplate* is macro-expanded before insertion into the message. If the *mflags* (surrounded by question marks) are specified, at least one of the specified flags must be stated in the mailer definition for this header to be automatically output. If one of these headers is in the input, it is reflected to the output, regardless of these flags.

Some headers have special semantics, described in the following sections.

### O - set option

There are a number of “random” options that can be set from a configuration file. Options are represented by single characters. The syntax of this line is:

**O** *o value*

This sets option *o* to be *value*. Depending on the option, *value* can be a string, an integer, a Boolean (with legal values “t”, “T”, “f”, or “F”; the default is TRUE), or a time interval.

### T - define trusted users

Trusted users are those who are permitted to override the sender address using the -f flag. These typically are *root*, *uucp*, and *network*, but in some cases it may be convenient to extend this list to include other users, perhaps to support a separate UUCP login for each host. The syntax of this line is:

**T** *user1 user2...*

There may be more than one of these lines.

### P - precedence definitions

Values for the “Precedence:” field may be defined using the P control line. The syntax of this field is:

**P** *name=num*

When the *name* is found in a “Precedence:” field, the message class is set to *num*. Higher numbers mean higher precedence. Numbers below zero have the special property that error messages cannot be returned. The default precedence is zero. For example, our list of precedences is:

```
Pfirst-class=0
Pspecial-delivery=100
Pjunk=-100
```

## The semantics

This section describes the semantics of the configuration file.

### Special macros, conditionals

Macros are interpolated using the construct *\$x*, where *x* is the name of the macro to be interpolated. In particular, lowercase letters are reserved to have special semantics, used to pass information in or out of *sendmail*; some special characters are reserved to provide conditionals; and so on.

Conditionals can be specified using the syntax:

**\$?***x text1 \$| text2 \$.*

This interpolates *text1* if the macro *\$x* is set, and *text2* otherwise. The “else” (*\$|*) clause can be omitted.

The following macros must be defined to transmit information into **sendmail**:

- e** SMTP entry message
- j** "official" domain name for this site
- l** format of the UNIX system from line
- n** name of the daemon (for error messages)
- o** set of "operators" in addresses
- q** default format of sender address

The **\$e** macro is printed out when SMTP starts up. The first word must be the **\$j** macro. The **\$j** macro should be in RFC821 format. The **\$l** and **\$n** macros can be considered constants, except under terribly unusual circumstances. The **\$o** macro consists of a list of characters that are considered tokens and that separate tokens when parsing. For example, if "@" were in the **\$o** macro, then the input **a@b** would be scanned as three tokens: "a", "@", and "b". Finally, the **\$q** macro specifies how an address should appear in a message when it is defaulted. For example, on our system, these definitions are:

```
De$j Sendmail $v ready at $b
DnMAILER-DAEMON
DlFrom $g $d
Do.:%@!^=/
Dqgx ($x)$.
Dj$H.$D
```

An acceptable alternative for the **\$q** macro is **\$?x\$x \$.<\$g>**. These correspond to the following two formats:

```
eric@Ocelot (Eric Allman)
Eric Allman <eric@Ocelot>
```

Some macros are defined by **sendmail** for interpolation into **argv**'s for mailers or for other contexts. These macros are:

- a** origination date in Arpanet format
- b** current date in Arpanet format
- c** hop count
- d** date in UNIX (ctime) format
- f** sender (from) address
- g** sender address relative to the recipient
- h** recipient host
- i** queue id
- p** **sendmail**'s pid
- r** protocol used
- s** sender's host name

- t**    numeric representation of the current time
- u**    recipient user
- v**    version number of sendmail
- w**    hostname of this site
- x**    full name of the sender
- z**    home directory of the recipient

There are three types of dates that can be used. The **\$a** and **\$b** macros are in Arpanet format; **\$a** is the time as extracted from the "Date:" line of the message (if there was one), and **\$b** is the current date and time (used for postmarks). If no "Date:" line is found in the incoming message, **\$a** is set to the current time also. The **\$d** macro is equivalent to the **\$a** macro in UNIX system (ctime) format.

The **\$f** macro is the ID of the sender as originally determined; when you are mailing to a specific host, the **\$g** macro is set to the address of the sender *relative to the recipient*. For example, if you send to *bollard@matisse* from the machine *ucbarpa*, the **\$f** macro is "eric" and the **\$g** macro is "eric@ucbarpa."

The **\$x** macro is set to the full name of the sender. This can be determined in several ways. It can be passed as a flag to **sendmail**. The second choice is the value of the "Full-name:" line in the header if it exists, and the third choice is the comment field of a "From:" line. If all of these fail, and if the message is being originated locally, the full name is looked up in the */etc/passwd* file.

When sending, the **\$h**, **\$u**, and **\$z** macros are set to the host, user, and home directories (if local) of the recipient. The first two are set from the **\$@** and **\$:** parts of the rewriting rules, respectively.

The **\$p** and **\$t** macros create unique strings (for example, for the "Message-Id:" field). The **\$i** macro is set to the queue ID on this host; if put into the timestamp line, it can be extremely useful for tracking messages. The **\$v** macro is set to be the version number of **sendmail**; this is normally put in timestamps and has proved extremely useful for debugging. The **\$w** macro is set to the name of this host, if it can be determined. The **\$c** field is set to the "hop count," that is, the number of times this message has been processed. This can be determined by the **-h** flag on the command line or by counting the timestamps in the message.

The **\$r** and **\$s** fields are set to the protocol used to communicate with **sendmail** and the sending *hostname*; these are not supported in the current version.

**Special classes**

Set the class `$=w` to be the set of all names by which this host is known. This can be used to delete local hostnames.

**The left-hand side**

The left-hand side of rewriting rules contains a pattern. Normal words are simply matched directly. Metasyntax is introduced using a dollar sign. The metasymsymbols are:

- `$*` match zero or more tokens
- `$+` match one or more tokens
- `$-` match exactly one token
- `$=x` match any token in class *x*
- `$~x` match any token not in class *x*

If any of these match, they are assigned to the symbol `$n` for replacement on the right-hand side, where *n* is the index in the LHS. For example:

`$-:$+`

Suppose the LHS is applied to the following input:

UCBARPA:eric

The rule matches, and the values passed to the RHS are:

`$1` UCBARPA  
`$2` eric

**The right-hand side**

When the left-hand side of a rewriting rule matches, the input is deleted and replaced by the right-hand side. Tokens are copied directly from the RHS, unless they begin with a dollar sign. Metasymsymbols are:

- `$n` substitutes indefinite token *n* from LHS
- `$(name$)` canonicalizes *name*
- `$>n` calls ruleset *n*
- `$#mailer` resolves to *mailer*
- `$@host` specifies *host*
- `$:user` specifies *user*

The `$n` syntax substitutes the corresponding value from a `$+`, `$-`, `$*`, `$=`, or `$~` match on the LHS. It can be used anywhere.

A host name enclosed between `$(` and `$)` is looked up using the `gethostent(3)` routines and replaced by the canonical name. For example, `$[[128.32.130.2]]$` would become `vangogh.berkeley.edu`, and `$(csam$)` would become `lbl-csam.arpa`.

The `$>n` syntax causes the remainder of the line to be substituted as usual and then passed as the argument to ruleset *n*. The final value of ruleset *n* then becomes the substitution for this rule.

The **\$\$** syntax should only be used in ruleset zero. It causes evaluation of the ruleset to terminate immediately, and it signals to **sendmail** that the address has completely resolved. The complete syntax is:

**\$\$mailer\$@host\$:user**

This specifies the {mailer, host, user} 3-tuple necessary to direct the mailer. If the mailer is local, the host part can be omitted. The *mailer* and *host* must be a single word, but the *user* can be multi-part.

A RHS can also be preceded by a **\$@** or a **\$:** to control evaluation. A **\$@** prefix causes the ruleset to return with the remainder of the RHS as the value. A **\$:** prefix causes the rule to terminate immediately, but the ruleset to continue. This can avoid continued application of a rule. The prefix is stripped before continuing.

The **\$@** and **\$:** prefixes can precede a **\$>** specification. For example, the form:

**R\$+     \$:\$>7\$1**

matches anything, passes that to ruleset seven, and continues; the **\$:** is necessary to avoid an infinite loop.

Substitution occurs in the order described; that is, parameters from the LHS are substituted, hostnames are canonicalized, "subroutines" are called and, finally, **\$\$**, **\$@**, and **\$:** are processed.

### Semantics of rewriting rule sets

There are five rewriting sets that have specific semantics. These are related as follows:

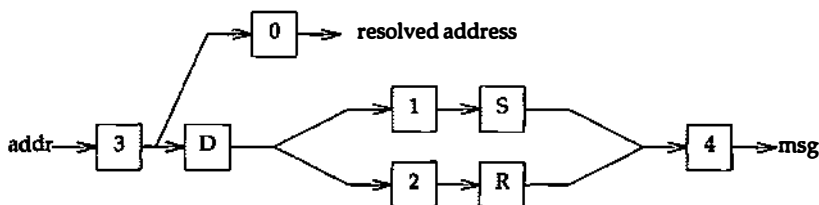


**Figure 15-1** Rewriting set semantics

D - sender domain addition

S - mailer-specific sender rewriting

R - mailer-specific recipient rewriting

Ruleset three should turn the address into "canonical form." This form should have the basic syntax:

**local-part@host-domain-spec**

If no “@” sign is specified, then the host-domain-spec can be appended from the sender address (if the C flag is set in the mailer definition corresponding to the sending mailer). Ruleset three is applied by **sendmail** before doing anything with any address.

Ruleset zero is applied after ruleset three to addresses that are actually going to specify recipients. It must resolve to a {mailer, host, user} triple. The mailer must be defined in the mailer definitions from the configuration file. The host is defined into the **\$h** macro for use in the **argv** expansion of the specified mailer.

Rulesets one and two are applied to all sender and recipient addresses, respectively. They are applied before any specification in the mailer definition. They must never resolve.

Ruleset four is applied to all addresses in the message. It is typically used to translate internal to external form.

### **Mailer flags**

There are several flags that can be associated with each mailer, each identified by a letter of the alphabet. Many of them are assigned semantics internally. Any other flags can freely assign headers conditionally to messages destined for particular mailers.

### ***The “error” mailer***

The mailer with the special name “error” can generate a user error. The (optional) host field is a numeric exit status to be returned, and the user field is a message to be printed. For example:

```
$#error$:Host unknown in this domain
```

The entry on the RHS of a rule causes the specified error to be generated if the LHS matches. This mailer is only functional in ruleset zero.

### ***Building a configuration file from scratch***

Building a configuration file from scratch is an extremely difficult job. Fortunately, it is almost never necessary to do so; nearly every situation that may come up may be resolved by changing an existing table. In any case, it is critical that you understand what you are trying to do and come up with a philosophy for the configuration table. This section is intended to explain the real purpose of a configuration table and to give you some ideas as to what your philosophy might be.

#### **Purpose of the configuration table**

The configuration table has three major purposes. The first and simplest is to set up the environment for **sendmail**. This involves setting the options, defining a few critical macros, and so on. Because these are described in other sections, we will not go into more detail here.

The second purpose is to rewrite addresses in the message. This should typically be done in two phases. The first phase maps addresses in any format into a canonical form. This should be done in ruleset three. The second phase maps this canonical form into the syntax appropriate for the receiving mailer.

The **sendmail** program performs this second phase in the following three sub-phases: Rulesets one and two are applied to all sender and recipient addresses, respectively. After this, you can specify per-mailer rulesets for both sender and recipient addresses. This allows mailer-specific customization. Finally, ruleset four is applied to do any default conversion to external form.

The third purpose of the configuration table is to map addresses into the actual set of instructions necessary to get the message delivered. Ruleset zero must resolve to the internal form, which is in turn used as a pointer to a mailer descriptor. This describes the interface requirements of the mailer.

### Relevant issues

The canonical form you use should almost certainly be as specified in the Arpanet standards documents RFC819 and RFC822. RFC822 describes the format of the mail message itself. The **sendmail** program follows this RFC closely, to the extent that many of the standards described in this document cannot be changed without changing the code. In particular, the following characters have special interpretations:

`<>() "\`

Any attempt to use these characters for other than their RFC822 purpose in addresses is probably doomed to disaster.

RFC819 describes the specifics of the domain-based addressing. This is touched on in RFC822 as well. Essentially, each host is given a name that is a right-to-left dot-qualified pseudo-path from a distinguished root. The elements of the path need not be physical hosts; the domain is logical rather than physical. For example, at Ocelot, one legal host might be "a.CC.Ocelot.EDU"; reading from right to left, "EDU" is a top level domain comprising educational institutions, "Ocelot" is a logical domain name, "CC" represents the Computer Center, (in this case a strictly logical entity), and "a" is a host in the Computer Center.

### How to proceed

Once you have decided on a philosophy, it is worth examining the available configuration tables to decide whether any of them are close enough for you to use major parts of them as a basis for your configuration files. Even under the worst of conditions, there should be a large amount of material you can use.



The next step is to build ruleset three. This is the hardest part of the job. Beware of doing too much to the address in this ruleset, because anything you do reflects through to the message. In particular, stripping of local domains is best deferred, as this can leave you with addresses with no domain specifications at all. Because *sendmail* likes to append the sending domain to addresses with no domain, this can change the semantics of addresses. Also, try to avoid fully qualifying domains in this ruleset. Although technically legal, this can lead to unpleasantly and unnecessarily long addresses reflected into messages. The Ocelot configuration files define ruleset nine to qualify domain names and strip local domains. This is called from ruleset zero to get all addresses into a cleaner form.

Once you have ruleset three finished, the other rulesets should be relatively trivial. If you need hints, examine the supplied configuration tables.

#### Testing the rewriting rules: the **-bt** flag

When you build a configuration table, you can do a certain amount of testing using the "test mode" of *sendmail*. For example, you could invoke *sendmail* as:

```
sendmail -bt -Ctest.cf
```

This reads the configuration file *test.cf* and enter test mode. In this mode, you enter lines of the form:

```
rwset address
```

Here *rwset* is the rewriting set you want to use and *address* is an address to which to apply the set. Test mode shows you the steps it takes as it proceeds, finally showing you the address with which it ends up. You may use a comma-separated list of *rwsets* for sequential application of rules to an input; ruleset three is always applied first. For example:

```
1,21,4 monet:bollard
```

The first applies ruleset three to the input "monet:bollard." Ruleset one is then applied to the output of ruleset three, followed similarly by rulesets 21 and 4.

If you need more detail, you can also use the **-d21** flag to turn on more debugging. For example:

```
sendmail -bt -d21.99
```

This turns on a huge amount of information. A single-word address probably prints out several pages of information.

#### Building mailer descriptions

To add an outgoing mailer to your mail system, you must define the characteristics of the mailer.

Each mailer must have an internal name. This can be arbitrary, except that the names "local" and "prog" must be defined. The pathname of the mailer must be given in the P field. If this mailer should be accessed via an IPC connection, use the string "[IPC]" instead.

The F field defines the mailer flags. You should specify an "f" or "r" flag to pass the name of the sender as a -f or -r flag, respectively. These flags are only passed if they were passed to `sendmail`, so that mailers that give errors under some circumstances can be placated. If the mailer is not picky, you can just specify "-f \$g" in the argv template. If the mailer must be called as `root`, the S flag should be given. This does not reset the user ID before calling the mailer. If this mailer is local (that is, it performs final delivery rather than another network hop), the l flag should be given. Quote characters (backslashes and " marks) can be stripped from addresses if the s flag is specified. If this is not given, they are passed through. If the mailer is capable of sending to more than one user on the same host in a single transaction, the m flag should be stated. If this flag is on, then the argv template containing \$u is repeated for each unique user on a given host. The e flag marks the mailer as being expensive, which causes `sendmail` to defer connection until a queue run.

An unusual case is the C flag. This flag applies to the mailer that the message is received from, rather than the mailer being sent to; if set, the domain specification of the sender (that is, the @host.domain part) is saved and is appended to any addresses in the message that do not already contain a domain specification. For example:

```
From: eric@ucbarpa
To: wnj@monet, mckee
```

A message of this form is modified to:

```
From: eric@ucbarpa
To: wnj@monet, mckee@ucbarpa
```

This happens if and only if the C flag is defined in the mailer corresponding to `eric@ucbarpa`.

The S and R fields in the mailer description are per-mailer rewriting sets to be applied to sender and recipient addresses, respectively. These are applied after the sending domain is appended and the general rewriting sets (numbers one and two) are applied, but before the output rewrite (ruleset four) is applied. A typical use is to append the current domain to addresses that do not already have a domain. For example:

```
From: eric
```

A header of this form might be changed to be:

```
From: eric@ucbarpa
```

Or to this form, depending on the domain it is being shipped into:

From: ucbvax!eric

These sets can also be used to do special-purpose output rewriting in cooperation with ruleset four.

The E field defines the string to use as an end-of-line indication. A string containing only newline is the default. The usual backslash escapes (`\r`, `\n`, `\f`, `\b`) may be used.

Finally, an argv template is given as the A field. It may have embedded spaces. If there is no argv with a `$u` macro in it, **sendmail** speaks SMTP to the mailer. If the pathname for this mailer is `[IPC]`, the argv should be:

IPC \$h [ *port* ]

Here *port* is the optional port number to connect to.

For example:

```
Mlocal, P=/usr/bin/mail, F=lsDFMmnpS S=10, R=20, A=lmail $u
Mether, P=[IPC], F=meC, S=11, R=21, A=IPC $h, M=100000
```

These specifications specify a mailer to do local delivery and a mailer for Ethernet delivery. The first is called local; it is located in the file `/bin/mail`, takes a picky `-r` flag, and does local delivery; quotes should be stripped from addresses, and multiple users can be delivered at once; ruleset 10 should be applied to sender addresses in the message, and ruleset 20 should be applied to recipient addresses. The argv to send to a message is the word "mail," the word "-d," and words containing the name of the receiving user. If a `-r` flag is inserted, it is between the words mail and -d. The second mailer is called ether; it should be connected via an IPC connection; it can handle multiple users at once; connections should be deferred; and any domain from the sender address should be appended to any receiver name without a domain. Sender addresses should be processed by ruleset 11 and recipient addresses by ruleset 21. There is a 100,000-byte limit on messages passed through this mailer.

## *Configuration options*

---

The following options can be set using the `-o` flag on the command line or the O line in the configuration file. Many of them cannot be specified unless the invoking user is trusted.

- Afile**      Use the named *file* as the alias file. If no file is specified, use *aliases* in the current directory.
- aN**        If set, wait up to *N* minutes for an `@: @` entry to exist in the alias database before starting up. If it does not appear in *N* minutes, rebuild the database (if the **D** option is also set) or issue a warning.

- Bc** Set the blank substitution character to *c*. Unquoted spaces in addresses are replaced by this character.
- c** If an outgoing mailer is marked as being expensive, do not connect immediately. This requires that queueing be compiled in, because it depends on a queue-run process to actually send the mail.
- dx** Deliver in mode *x*. Legal modes are:
  - i** Deliver interactively (synchronously).
  - b** Deliver in background (asynchronously).
  - q** Just queue the message (deliver during queue run).
- D** If set, rebuild the alias database if necessary and possible. If this option is not set, **sendmail** will never rebuild the alias database unless explicitly requested using **-bi**.
- ex** Dispose of errors using mode *x*. The values for *x* are:
  - p** Print error messages (default).
  - q** No messages, just give exit status.
  - m** Mail back errors.
  - w** Write back errors (mail if user not logged in).
  - e** Mail back errors and give zero exit status always.
- Fn** This is the temporary file mode, in octal. 644 and 600 are good choices.
- f** Save UNIX-style "From" lines at the front of headers. Normally, they are assumed redundant and discarded.
- gn** Set the default group ID for mailers to run into *n*.
- Hfile** Specify the help file for SMTP.
- I** Use the domain name server, **named**.
- i** Ignore dots in incoming messages.
- Ln** Set the default log level to *n*.
- Mxvalue** Set the macro *x* to *value*. This is intended only for use from the command line.
- m** Send to me, too, even if I am in an alias expansion.
- Nnetname** The name of the home network (ARPA is the default). The argument of an SMTP HELO command is checked against *hostname.netname*, where *hostname* is requested from the kernel for the current connection. If they do not match, Received: lines are augmented by the name that is determined in this manner, so that messages can be traced accurately.
- o** Assume that the headers may be in old format; that is, spaces delimit names. This actually turns on an adaptive algorithm: if any

recipient address contains a comma, parentheses, or angle bracket, it is assumed that commas already exist. If this flag is not on, only commas delimit names. Headers are always output with commas between the names.

|                |                                                                                                                                                                                                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Qdir</b>    | Use the named <i>dir</i> as the queue directory.                                                                                                                                                                                                                                                                    |
| <b>qfactor</b> | Use <i>factor</i> as the multiplier in the map function to decide when just to queue up jobs rather than run them. This value is divided by the difference between the current load average and the load average limit ( <i>x</i> flag) to determine the maximum message priority that is sent. Defaults to 10,000. |
| <b>rtime</b>   | Timeout reads after <i>time</i> interval.                                                                                                                                                                                                                                                                           |
| <b>Sfile</b>   | Log statistics in the named <i>file</i> .                                                                                                                                                                                                                                                                           |
| <b>s</b>       | Be extra safe when running things, that is, always instantiate the queue file, even if you are going to attempt immediate delivery. The <b>sendmail</b> program always instantiates the queue file before returning control to the client.                                                                          |
| <b>Ttime</b>   | Set the queue timeout to <i>time</i> . After this interval, messages that have not been successfully sent is returned to the sender.                                                                                                                                                                                |
| <b>tS,D</b>    | Set the local time zone name to <i>S</i> for standard time and <i>D</i> for daylight time. This is only used under version six.                                                                                                                                                                                     |
| <b>un</b>      | Set the default user ID for mailers to <i>n</i> . Any mailer without the <i>S</i> flag in the mailer definition runs as this user.                                                                                                                                                                                  |
| <b>v</b>       | Run in verbose mode.                                                                                                                                                                                                                                                                                                |
| <b>w</b>       | Allow wildcard MX records.                                                                                                                                                                                                                                                                                          |
| <b>xLA</b>     | When the system-load average exceeds <i>LA</i> , just queue messages (that is, do not try to send them).                                                                                                                                                                                                            |
| <b>XLA</b>     | When the system load average exceeds <i>LA</i> , refuse incoming SMTP connections.                                                                                                                                                                                                                                  |
| <b>yfact</b>   | The indicated <i>factor</i> is added to the priority (thus <i>lowering</i> the priority of the job) for each recipient, that is, this value penalizes jobs with large numbers of recipients.                                                                                                                        |
| <b>Y</b>       | If set, deliver each job that is run from the queue in a separate process. Use this option if you are short of memory, because the default tends to consume considerable amounts of memory while the queue is being processed.                                                                                      |
| <b>zfact</b>   | The indicated <i>factor</i> is multiplied by the message class (determined by the "Precedence:" field in the user header and the <b>P</b> lines in the configuration file) and subtracted from the priority. Thus, messages with a higher Priority is favored.                                                      |

**Zfact** The *factor* is added to the priority every time a job is processed. Thus, each time a job is processed, its priority is decreased by the indicated value. In most environments, this should be positive, because hosts that are down are all too often down for a long time.

## Running sendmail

---

The following sections illustrate the flags and options used when executing **sendmail**, which is invoked from the */etc/tcp* shell script at startup. The *tcp* script is initialized by the **mkdev sendmail-init** and **mkdev cf** scripts, which are described earlier in this chapter. Refer to *tcp(ADMN)* for more information about */etc/tcp*.

### Command line flags

---

Arguments must be presented with flags before addresses. The flags are:

- f *addr*** The sender's machine address is *addr*. This flag is ignored unless the real user is listed as a "trusted user" or *addr* contains an exclamation point (because of certain restrictions in UUCP).
- r *addr*** This flag is an obsolete form of **-f**.
- h *cnt*** This flag sets the "hop count" to *cnt*. This represents the number of times this message has been processed by **sendmail** (to the extent that it is supported by the underlying networks). *cnt* is incremented during processing, and if it reaches MAXHOP (currently 30) **sendmail** throws away the message with an error.
- F*name*** This flag sets the full name of this user to *name*.
- n** Do not alias or forward.
- t** Read the header for To:, Cc:, and Bcc: lines, and send to everyone in those lists. The Bcc: line is deleted before sending. Any addresses in the argument vector is deleted from the send list.
- bx** Set operation mode to *x*. The operation modes are:
  - m** Deliver mail (default).
  - a** Run in ARPANET mode (see below).
  - s** Speak SMTP on input side.
  - d** Run as a daemon.
  - t** Run in test mode.
  - v** Just verify addresses, do not collect or deliver.
  - i** Initialize the alias database.
  - p** Print the mail queue.
  - z** Freeze the configuration file.

The special processing for the ARPANET includes reading the From: line from the header to find the sender, printing ARPANET-style messages (preceded by three-digit reply codes for compatibility with the FTP protocol), and ending lines of error messages with <CRLF>.

- qtime** Try to process the queued up mail. If the time is given, **sendmail** runs through the queue at the specified interval to deliver queued mail; otherwise, it only runs once.
- Cfile** Use a different configuration file. The **sendmail** program runs as the invoking user (rather than *root*) when this flag is specified.
- dlevel** Set debugging level.
- oxvalue** Set option *x* to the specified *value*. These options are described in the next section.

There are a number of options that can be specified as primitive flags (provided for compatibility with **delivermail**). These are the **e**, **i**, **m**, and **v** options. Also, the **f** option can be specified as the **-s** flag.

## ***Mailer flags***

---

The following flags can be set in the mailer description.

- f** The mailer wants a **-f from** flag, but only if this is a network forward operation (that is, the mailer gives an error if the executing user does not have special permissions).
- r** This is the same as **f**, but sends a **-r** flag.
- S** Do not reset the user ID before calling the mailer. This would be used in a secure environment where **sendmail** ran as *root*. This could be used to avoid forged addresses. This flag is suppressed if given from an unsafe environment (for example, a user's *mail.cf* file).
- n** Do not insert a UNIX-system-style From: line on the front of the message.
- l** This mailer is local (that is, final delivery is performed).
- s** Strip quote characters off the address before calling the mailer.
- m** This mailer can send to multiple users on the same host in one transaction. When a **\$u** macro occurs in the *argv* part of the mailer definition, that field is repeated as necessary for all qualifying users.
- F** This mailer wants a From: header line.

- D This mailer wants a Date: header line.
- M This mailer wants a Message-Id: header line.
- x This mailer wants a Full-Name: header line.
- P This mailer wants a Return-Path: line.
- u Uppercase should be preserved in user names for this mailer.
- h Uppercase should be preserved in host names for this mailer.
- A This is an Arpanet-compatible mailer, and all appropriate modes should be set.
- U This mailer wants UNIX-system-style From lines with the UUCP-style "remote from <host>" on the end.
- e This mailer is expensive to connect to, so try to avoid connecting normally. Any necessary connection occurs during a queue run.
- X This mailer wants to use the hidden dot algorithm as specified in RFC821. Basically, any line beginning with a dot has an extra dot prepended (to be stripped at the other end). This ensures that lines in the message containing a dot does not terminate the message prematurely.
- L Limit the line lengths as specified in RFC821.
- P Use the return-path in the SMTP "MAIL FROM:" command, rather than just the return address. Although this is required in RFC821, many hosts do not process return paths properly.
- I This mailer is speaking SMTP to another *sendmail*. As such, it can use special protocol features. This option is not required (that is, if this option is omitted, the transmission still operates successfully, although perhaps not as efficiently as possible).
- C If mail is received from a mailer with this flag set, any addresses in the header that do not have an at sign "@" after being rewritten by ruleset three has the @domain clause from the sender tacked on. This allows mail with headers of the form to be rewritten automatically:
 

From: usera@hosta  
 To: userb@hostb, userc

It becomes:

From: usera@hosta  
 To: userb@hostb, userc@hosta
- E Escape lines beginning with "From" in the message with a ">" sign.



## Arguments

---

Some important arguments of the **sendmail** program are described here.

### *Queue interval*

The amount of time between forking a process to run through the queue is defined by the **-q** flag. If you run in mode **f** or **a**, this can be relatively large, because it is only relevant when a host that was down comes back up. If you run in **q** mode, it should be relatively short, because it defines the maximum amount of time that a message may sit in the queue.

### *Daemon mode*

If you allow incoming mail over an IPC connection, you should have a daemon running. This should be set by your */etc/rc* file using the **-bd** flag. The **-bd** flag and the **-q** flag may be combined in one call:

```
/usr/lib/sendmail -bd -q30m
```

### *Forcing the queue*

In some cases, you may find that the queue has gotten clogged for some reason. You can force a queue run using the **-q** flag (with no value). It is entertaining to use the **-v** flag (verbose) when this is done, to watch what happens:

```
/usr/lib/sendmail -q -v
```

### *Debugging*

There is a fairly large number of debug flags built into **sendmail**. Each debug flag has a number and a level, where higher levels cause more information to be printed out. The convention is that levels greater than nine are not required. They print out so much information that you would not normally want to see them, except for debugging that particular piece of code. Debug flags are set using the **-d** option. The syntax is:

```
debug-flag: -d debug-list
debug-list: debug-option [, debug-option]
debug-option: debug-range [. debug-level]
debug-range: integer | integer - integer
debug-level: integer
```

The spaces are for reading ease only. For example:

```
-d12 Set flag 12 to level 1.
-d12.3 Set flag 12 to level 3.
-d3-17 Set flags 3 through 17 to level 1.
-d3-17.4 Set flags 3 through 17 to level 4.
```

## *Trying a different configuration file*

An alternative configuration file can be specified using the **-C** flag. The following example uses the configuration file *test.cf* instead of the default */usr/lib/sendmail.cf*.

```
/usr/lib/sendmail -Ctest.cf
```

If the **-C** flag has no value, it defaults to *sendmail.cf* in the current directory.

## *Changing the values of options*

Options can be overridden using the **-o** flag. For example:

```
/usr/lib/sendmail -oT2m
```

This sets the **T** (timeout) option to two minutes for this run only.

## *Tuning*

---

There are a number of configuration parameters you may want to change, depending on the requirements of your site. Most of these are set using an option in the configuration file. For example, the line "OT3d" sets option "T" to the value "3d" (three days).

Most of these options default appropriately for most sites. However, sites having very high mail loads may find they need to tune them as appropriate for their mail load. In particular, sites experiencing a large number of small messages, many of which are delivered to many recipients, may find that they need to adjust the parameters dealing with queue priorities.

### *Timeouts*

All time intervals are set using a scaled syntax. For example, "10m" represents ten minutes, whereas "2h30m" represents two-and-a-half hours. The full set of scales is:

```
s seconds
m minutes
h hours
d days
w weeks
```

### *Queue interval*

The argument to the **-q** flag specifies how often a subdaemon runs the queue. This is typically set to between fifteen minutes and one hour.

### *Read timeouts*

It is possible to time out when reading the standard input or when reading from a remote SMTP server. Technically, this is not acceptable within the published protocols. However, it might be appropriate to set it to something large (such as an hour) in certain environments. This reduces the chance of

large numbers of idle daemons piling up on your system. This timeout is set using the `r` option in the configuration file.

### **Message timeouts**

After sitting in the queue for a few days, a message times out. This means that if a message cannot be delivered for some reason, it is returned to the sender. This ensures that at least the sender is aware that the message was not sent. The timeout is typically set to three days. This timeout is set using the `T` option in the configuration file.

The time of submission is set in the queue, rather than the amount of time left until timeout. As a result, you can flush messages that have been hanging for a short period by running the queue with a short message timeout. For example:

```
/usr/lib/sendmail -oT1d -q
```

This runs the queue and flushes anything that is one day old.

### ***Forking during queue runs***

When the `Y` option is set, **sendmail** forks before each individual message while running the queue. This prevents **sendmail** from consuming large amounts of memory, and so it may be useful in memory-poor environments. However, if the `Y` option is not set, **sendmail** keeps track of hosts that are down during a queue run, which can improve performance dramatically.

### ***Queue priorities***

Every message is assigned a priority when it is first instantiated, consisting of the message size (in bytes) offset by the message class multiplied by the "work class factor" and the number of recipients multiplied by the "work recipient factor." The priority plus the creation time of the message (in seconds since January 1, 1970) are used to order the queue. Higher numbers for the priority mean that the message is processed later, when running the queue.

The message size is included so that large messages are penalized relative to small messages. The message class allows users to send high-priority messages by including a "Precedence:" field in their message; the value of this field is looked up in the `P` lines of the configuration file. Because the number of recipients affects the size of load a message presents to the system, this is also included in the priority.

The recipient and class factors can be set in the configuration file by using the `y` and `z` options, respectively. They default to 1000 (for the recipient factor) and 1800 (for the class factor). The initial priority is:

```
pri = size - (class * z) + (nrcpt * y)
```

(Remember, higher values for this parameter actually mean that the job is treated with lower priority.)

The priority of a job can also be adjusted each time it is processed (that is, each time an attempt is made to deliver it) using the “work time factor,” set by the Z option. This is added to the priority, so it normally decreases the precedence of the job, on the grounds that jobs that have failed many times tend to fail again in the future.

## *Delivery mode*

There are a number of delivery modes for **sendmail**, and they are set by the d configuration option. These modes specify how quickly mail is delivered. Legal modes are:

- i deliver interactively (synchronously)
- b deliver in background (asynchronously)
- q queue only (do not deliver)

There are tradeoffs. Mode “i” passes the maximum amount of information to the sender, but it is hardly ever necessary. Mode “q” puts the minimum load on your machine, but means that delivery may be delayed for up to the queue interval. Mode “b” is probably a good compromise. However, this mode can cause large numbers of processes if you have a mailer that takes a long time to deliver a message.

## *File modes*

There are several files involved with **sendmail** that can have a number of modes. The modes depend on the functionality you want and the level of security you require.

### **To suid or not to suid?**

The **sendmail** program can safely be made setuid to *root*. At the point where it is about to **exec(S)** a mailer, it checks to see if the user ID is zero. If so, it resets the user ID and groupid to a default (set by the u and g options). (This can be overridden by setting the S flag to the mailer for mailers that are trusted and must be called as *root*.) However, this causes mail processing to be accounted to *root* rather than to the user sending the mail.

### **Temporary file modes**

The mode of all temporary files that **sendmail** creates is determined by the F option. Reasonable values for this option are 0600 and 0644. If the more permissive mode is selected, it is not necessary to run **sendmail** as *root* at all (even when running the queue).

### **Should the alias database be writable?**

The database that **sendmail** actually uses is represented by two files. These files are:

- *aliases.dir*
- *aliases.pag*

Both files are located in */usr/lib/mail*. The mode on these files should match the mode on */usr/lib/mail/aliases*. If *aliases* is writable and the DBM files (*aliases.dir* and *aliases.pag*) are not, users cannot reflect their desired changes through to the actual database. However, if *aliases* is read-only and the DBM files are writable, a slightly sophisticated user can arrange to steal mail anyway.

If your DBM files are not writable by the world or you do not have auto-rebuild enabled (with the **D** option), then you must be careful to reconstruct the alias database each time you change the text version via the command:

**newaliases**

If this step is ignored or forgotten, any intended changes are also ignored or forgotten.

## *Administering sendmail*

---

System administration with **sendmail** consists of monitoring the system log file, managing the mail queue, maintaining the system alias file, and performing other related actions.

### *System log*

---

The system log is entered in the file */usr/adm/syslog*. Each line in the system log consists of a timestamp, the name of the machine that generated it (for logging from several machines over the network), the word “sendmail,” and a message.

A large amount of information can be logged. The log is arranged as a succession of levels. At the lowest level, only extremely strange situations are logged. At the highest level, even the most mundane and uninteresting events are recorded for posterity. As a convention, log levels under 10 are considered “useful”; log levels above 10 are usually for debugging purposes.

### *The mail queue*

---

The mail queue should be processed transparently. However, you may find that manual intervention is sometimes necessary. For example, if a major host is down for a period of time the queue may become clogged. Although **sendmail** ought to recover gracefully when the host comes up, you may find performance unacceptably bad in the meantime.

The contents of the queue can be printed using the **mailq** command (or by specifying the **-bp** flag to **sendmail**):

**mailq**

This produces a listing of the queue identifiers, the size of the message, the date the message entered the queue, and the sender and recipients.

## *Format of sendmail queue files*

All queue files have the form *x*fAA99999, where AA99999 is the ID for this file and *x* is a type. The types are:

- d data file. The message body (excluding the header) is kept in this file.
- l lock file. If this file exists, the job is currently being processed, and a queue run does not process the file. For that reason, an extraneous *lf* file can cause a job to seem to disappear. It will not even time out.
- n This file is created when an ID is being created. It is a separate file to ensure that no mail can ever be destroyed due to a race condition. It should exist for no more than a few milliseconds at any given time.
- q queue control file. This file contains the information necessary to process the job.
- t temporary file. This is an image of the *qf* file when it is being rebuilt. It should be renamed to a *qf* file very quickly.
- x transcript file, existing during the life of a session, showing everything that happens during that session

The *qf* file is structured as a series of lines, each beginning with a code letter. The lines are as follows:

- D name of the data file. There can only be one of these lines.
- H header definition. There can be any number of these lines. The order is important: it represents the order in the final message. This uses the same syntax as header definitions in the configuration file.
- R recipient address. This is normally completely aliased, but is actually re-aliased when the job is processed. There is one line for each recipient.
- S sender address. There can only be one of these lines.
- E error address. If any such lines exist, they represent the addresses that should receive error messages.
- T job creation time. This computes how long a job remains in the queue undelivered, before being returned to the sender.
- P current message priority. This orders the queue. Higher numbers mean lower priorities. The priority changes as the message sits in the queue. The initial priority depends on the message class and the size of the message.
- M message. This line is printed by the **mailq** command, and is generally used to store status information. It can contain any text.

As an example, the following is a queue file sent to "mckee@calder" and "wnj:"

```
DdfA13557
Seric
T404261372
P132
Rmckee@calder
Rwnj
H?D?date: 23-Oct-82 15:49:32-PDT (Sat)
H?F?from: eric (Eric Allman)
H?x?full-name: Eric Allman
Hsubject: this is an example message
Hmessage-id: <8209232249.13557@UCBARPA.BERKELEY.ARPA>
Hreceived: by UCBARPA.BERKELEY.ARPA (3.227 [10/22/82])
 id A13557; 23-Oct-82 15:49:32-PDT (Sat)
HTo: mckee@calder, wnj
```

This shows the name of the data file, the person who sent the message, the submission time (in seconds since January 1, 1970), the message priority, the message class, the recipients, and the headers for the message.

### *Forcing the queue*

The **sendmail** program should run the queue automatically at intervals. The algorithm is to read and sort the queue, then to attempt to process all jobs in order. When it attempts to run the job, **sendmail** first checks to see if the job is locked. If so, it ignores the job.

There is no attempt to ensure that only one queue processor exists at any time, because there is no guarantee that a job cannot take forever to process. Due to the locking algorithm, it is impossible for one job to freeze the queue. However, an uncooperative recipient host or a program recipient that never returns can accumulate many processes in your system. Unfortunately, there is no way to resolve this without violating the protocol.

In some cases, you may find that if a major host goes down for a couple of days, this can create a prohibitively large queue. This situation causes **sendmail** to spend an inordinate amount of time sorting the queue. This situation can be fixed by moving the queue to a temporary place and creating a new queue. The old queue can be run later, when the offending host returns to service.

To do this, it is acceptable to move the entire queue directory:

```
cd /usr/spool
mv mqueue omqueue; mkdir mqueue; chmod 777 mqueue
```

You should then kill the existing daemon (because it is still processing in the old queue directory) and create a new daemon.

To run the old mail queue, run the following command:

```
/usr/lib/sendmail -oQ/usr/spool/omqueue -q
```

The **-oQ** flag specifies an alternate queue directory, and the **-q** flag says just to run every job in the queue. Use the **-v** flag to view what is going on.

When the queue is finally emptied, you can remove the directory:

```
rmdir /usr/spool/omqueue
```

## ***sendmail configuration file***

---

Almost all configuration information for **sendmail** is read at runtime from the ASCII file */usr/lib/sendmail.cf*. This file has macro definitions encoded in it. These define such details as the value of macros used internally, header declarations, mailer definitions and address rewriting rules.

The **sendmail** configuration file is created as shown earlier in this chapter in the section “Configuring sendmail.”

The header declarations tell **sendmail** the format of header lines that are processed specially. For example, any lines that are added or reformatted receive special processing. The mailer definitions give information such as the location and characteristics of each mailer. The address rewriting rules enable **sendmail** to be highly configurable and customizable, though this comes at the cost of some complexity.

## ***The alias database***

---

The alias database exists in two forms. One is a text form, maintained in the file */usr/lib/mail/aliases*. The aliases are of the form:

```
name: name1, name2, ...
```

Only local names can be aliased. For example:

```
eric@mit-xx: eric@berkeley.EDU
```

This does not have the desired effect. Aliases can be continued by starting any continuation line with a space or a tab. Blank lines and lines beginning with a numbersign “#” are comments.

The second form is processed by the **dbm(5)** library. This form is in the files */usr/lib/mail/aliases.dir* and */usr/lib/mail/aliases.pag*. This is the form that **sendmail** actually uses to resolve aliases. This technique improves performance.

### ***Rebuilding the alias database***

The DBM version of the database can be rebuilt explicitly by executing the command:

```
newaliases
```



This is equivalent to giving **sendmail** the **-bi** flag:

**/usr/lib/sendmail -bi**

If the “**D**” option is specified in the configuration, **sendmail** rebuilds the alias database automatically, if possible, when it is out of date. It does this under either or both of the following conditions:

- The DBM version of the database is mode 666.
- **sendmail** is running setuid to *root*.

Auto-rebuild can be dangerous on heavily loaded machines with large alias files; if it might take more than five minutes to rebuild the database, there is a chance that several processes start the rebuild process simultaneously.

### *Potential alias database problems*

There are a number of problems that can occur with the alias database. They all result when a **sendmail** process accesses the DBM version while it is only partially built. This can happen under two circumstances: either one process accesses the database while another process is rebuilding it, or the process rebuilding the database dies (due to being killed or a system crash) before completing the rebuild.

The **sendmail** program includes two techniques to try to relieve these problems. First, it ignores interrupts while rebuilding the database; this avoids the problem of someone aborting the process and leaving a partially rebuilt database. Second, at the end of the rebuild it adds an alias of the form:

@: @

(Note that this is not normally legal.) Before **sendmail** accesses the database, it checks to ensure that this entry exists. The **sendmail** program waits for this entry to appear, at which point it forces a rebuild itself.

### *List owners*

If an error occurs on sending to a certain address, say “*x*,” **sendmail** looks for an alias of the form “owner-*x*” to receive the errors. This is typically useful for a mailing list where the submitter of the list has no control over the maintenance of the list itself; in this case the list maintainer would be the owner of the list. For example:

```
unix-wizards: eric@ucbarpa, wnj@monet, nosuchuser,
 sam@matisse
owner-unix-wizards: eric@ucbarpa
```

This would cause “eric@ucbarpa” to get the error that occurs when someone sends to unix-wizards, due to the inclusion of “nosuchuser” on the list.

## Per-user forwarding (*forward* files)

---

As an alternative to the alias database, any user can put a file with the name *forward* in his or her home directory. If this file exists, **sendmail** redirects mail for that user to the list of addresses listed in the *forward* file. For example, suppose the home directory for user *mckee* has a *forward* file with contents:

```
mckee@ernie
kirk@calder
```

Then any mail arriving for *mckee* is redirected to the specified accounts.

## Special header lines

---

Several header lines have special interpretations defined by the configuration file. Others have interpretations built into **sendmail** that cannot be changed without changing the code. These built-ins are described here.

### *Return-receipt-to:*

If this header is sent, a message is sent to any specified addresses when the final delivery is completed, that is, when successfully delivered to a mailer with the **l** flag (local delivery) set in the mailer descriptor.

### *Errors-to:*

If errors occur anywhere during processing, this header causes error messages to go to the listed addresses rather than to the sender. This is intended for mailing lists.

### *Apparently-to:*

If a message comes in with no recipients listed in the message (in a **To:**, **Cc:**, or **Bcc:** line), then **sendmail** adds an "Apparently-To:" header line for any recipients it is aware of. This is not put in as a standard recipient line to warn any recipients that the list is not complete.

## Summary of support files

---

This is a summary of the support files that **sendmail** creates or generates.

|                            |                                                                                                                                                                          |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>/usr/lib/sendmail</i>   | binary of <b>sendmail</b>                                                                                                                                                |
| <i>/usr/bin/newaliases</i> | link to <i>/usr/lib/sendmail</i> ; causes the alias database to be rebuilt. Running this program is completely equivalent to giving <b>sendmail</b> the <b>-bi</b> flag. |
| <i>/usr/bin/mailq</i>      | prints a listing of the mail queue. This program is equivalent to using the <b>-bp</b> flag to <b>sendmail</b> .                                                         |

|                                        |                                                                           |
|----------------------------------------|---------------------------------------------------------------------------|
| <i>/usr/lib/sendmail.cf</i>            | configuration file, in textual form                                       |
| <i>/usr/lib/sendmail.fc</i>            | configuration file represented as a memory image                          |
| <i>/usr/lib/sendmail.hf</i>            | SMTP help file                                                            |
| <i>/usr/lib/sendmail.st</i>            | statistics file; need not be present                                      |
| <i>/usr/lib/mail/aliases</i>           | textual version of the alias file                                         |
| <i>/usr/lib/mail/aliases.{pag,dir}</i> | alias file in <b>dbm(S)</b> format                                        |
| <i>/usr/spool/mqueue</i>               | directory in which the mail queue and temporary files reside              |
| <i>/usr/spool/mqueue/qf*</i>           | control (queue) files for messages                                        |
| <i>/usr/spool/mqueue/df*</i>           | data files                                                                |
| <i>/usr/spool/mqueue/lf*</i>           | lock files                                                                |
| <i>/usr/spool/mqueue/tf*</i>           | temporary versions of the <i>qf</i> files, used during queue-file rebuild |
| <i>/usr/spool/mqueue/nf*</i>           | file used when creating a unique ID                                       |
| <i>/usr/spool/mqueue/hx*</i>           | transcript of the current session                                         |

## More *sendmail* information

---

The following manual pages provide additional information about **sendmail**:

| Manual page             | Description                                     |
|-------------------------|-------------------------------------------------|
| <b>aliases(SFF)</b>     | aliases file for <b>sendmail</b>                |
| <b>lmail(ADMN)</b>      | handle local mail delivery from <b>sendmail</b> |
| <b>mailaddr(ADMN)</b>   | mail addressing description                     |
| <b>mconnect(ADMN)</b>   | connect to SMTP mail server socket              |
| <b>rmail(ADMN)</b>      | handle remote mail received by UUCP             |
| <b>sendmail(ADMN)</b>   | <b>sendmail</b> command information             |
| <b>sendmail.cf(SFF)</b> | <b>sendmail</b> configuration file              |

## Chapter 16

# Using modems

---

Modems (from *modulate demodulate*) are a significant addition to your system, allowing you to communicate over phone lines from remote sites.

This chapter explains the following tasks:

- choosing a serial port
- installing and configuring modems
- dialing into your computer
- adding passwords for dial-in lines
- dialing out from your computer
- using and customizing dialers

## Choosing a serial port

---

Note that physical connections between a device and the system vary according to hardware configuration. For specific information about connecting your serial device, refer to the hardware manuals provided with the device and with your computer. Before adding a modem, you should make certain you have a port available, either directly on COM1 or COM2, or from a multiport card.

Table 16-1 contains sample device names of serial ports with and without modem control.

**Table 16-1 Serial ports**

| Device                  | Function                                       |
|-------------------------|------------------------------------------------|
| <code>/dev/tty1a</code> | main serial adapter without modem control      |
| <code>/dev/tty1A</code> | main serial adapter with modem control         |
| <code>/dev/tty2a</code> | alternate serial adapter without modem control |
| <code>/dev/tty2A</code> | alternate serial adapter with modem control    |

`/dev/tty1a` and `/dev/tty1A` refer to the same serial port (likewise for `/dev/tty2a` and `/dev/tty2A`). The operating system uses different device-driver subroutines for each. Never attempt to use both modem and non-modem control ports at the same time or you will see the warning:

cannot open: device busy

For systems with non-intelligent multiport serial cards, `/dev/tty[1,2][a-m]` are the non-modem control devices, and `/dev/tty[1,2][A-M]` are the modem control devices.

Intelligent multiport serial cards are supplied with their own drivers and have variety of device naming schemes; consult the documentation for your intelligent card for a list of device names.

For information on adding serial ports, see the *Hardware Configuration Guide*.

## Configuring your modem

Proper modem configuration is necessary when using `cu(C)` and `uucp(C)`. Modem settings differ for each modem. Consult your modem manual for the proper switch settings.

### Smartmodem 1200 or compatible

If you have a Hayes Smartmodem 1200 or compatible, switches 3 and 8 should be down:

|      | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|---|---|---|---|---|---|---|
| up   | • | • |   | • | • | • | • |   |
| down |   |   | • |   |   |   |   | • |

When switch 3 is down, the result codes are sent by the modem to the terminal or computer. When switch 8 is down, the modem can interpret the command being issued. This allows both MS-DOS and UNIX system communications systems to work.

Table 16-2 lists the functions of each switch setting.

**Table 16-2 Hayes-Compatible switch settings**

| Switch | Position | Function                                                      |
|--------|----------|---------------------------------------------------------------|
| 1      | up*      | Modem responds to DTR from computer                           |
|        | down     | Modem forces DTR high, so no signal is required from computer |
| 2      | up*      | Result codes are in English                                   |
|        | down     | Result codes are numeric                                      |
| 3      | up       | There are no result codes                                     |
|        | down*    | Result codes are sent in response to each modem command       |
| 4      | up*      | Commands are echoed                                           |
|        | down     | Commands are not echoed                                       |
| 5      | up*      | Modem answers phone                                           |
|        | down     | Modem does not answer phone                                   |
| 6      | up*      | CD is asserted when carrier is actually present               |
|        | down     | CD and DSR are forced high                                    |
| 7      | up*      | Modem is attached to single-line phone                        |
|        | down     | Modem is attached to multi-line phone                         |
| 8      | up       | Modem does not recognize dialing commands                     |
|        | down*    | Modem recognizes dialing commands                             |

The asterisks (\*) indicate the switch settings required. If you have a different modem, consult your reference manual for the proper switch settings to both send and receive calls.

### *Smartmodem 2400, V-series 9600 or compatible*

The Hayes 2400 and V-series 9600 Smartmodem or compatible modems are supported and are configured automatically when you use the dialer entries or dialer programs supplied with your SCO distribution. If you are using an unsupported dialer (one brought from elsewhere or written yourself), and you plan to use it for a dial-in line, be sure that you use the following settings:

|                  |                                                            |
|------------------|------------------------------------------------------------|
| <b>AT&amp;f</b>  | fetches factory configuration.                             |
| <b>ATT</b>       | is tone dialing.                                           |
| <b>ATl0</b>      | is low speaker volume.                                     |
| <b>AT&amp;d2</b> | sets dtr "2": goes on hook when dtr drops.                 |
| <b>AT&amp;c1</b> | sets dcd "1": dcd tracks remote carrier.                   |
| <b>ATs0=1</b>    | answers phone after 1 ring (AA light should come on).      |
| <b>ATs2=128</b>  | disables modem escape sequence.                            |
| <b>ATe0</b>      | does not echo (modem no longer echoes what is sent to it). |

- ATq1** is quiet mode (modem does not respond with “OK” after this command or any that follow).
- AT&w** saves settings in non-volatile memory. If you do not want to save the settings, you do not have to enter this command.

### Telebit Trailblazer

---

If you have a Telebit Trailblazer modem or compatible, log in as *root* and enter the following command:

```
/usr/lib/uucp/dialTBIT -z /dev/tty nn 9600
```

where *nn* is the tty number of the serial line.

### General modem configuration guidelines

---

When configuring non-supported modems, there are a number of things you should look out for:

- You may need to disable or reduce the level of data compression and error correction when using a modem for UUCP. If throughput seems too low, or you have failed transfers, try disabling these features.
- Do not enable XON/XOFF flow control when using the modem for UUCP. The stop characters (DC3) can occur in the UUCP protocol data stream, and thus cause transfer failures. Make sure that XON/XOFF flow control is turned off (XON/XOFF passthrough mode may be usable).

You can use RTS/CTS flow control with your modem; see “RTS/CTS flow control” (page 840) for more information.

If you want to enable XON/XOFF when using the modem for *cu(C)*, then you will need to have an alternate dialer defined. This can be done by modifying and renaming the *Dialers* file entry, or by modifying and renaming the dialer program, such that you have one dialer definition for UUCP and another for *cu*. See the *sysfiles(F)* manual page for instructions on how to do this.

### Installing a modem

---

**NOTE** We strongly advise that you do not use an internal modem. We have successfully tested the Hayes 1200 and 2400 baud internal modems, and these modems are known to work. However, most Hayes-compatible internal modems have compatibility problems which prevent them from working properly under UNIX System V/386. These incompatibilities include:

- No support for quiet mode, no-echo mode, or auto-answer mode.
- Modem floods the bus with spurious interrupts.

When you are hooking up your modem, or any other device, make sure that serial wires connected to your computer are not left hanging. An

## Installing a modem

---

**NOTE** We strongly advise that you do not use an internal modem. We have successfully tested the Hayes 1200 and 2400 baud internal modems, and these modems are known to work. However, most Hayes-compatible internal modems have compatibility problems which prevent them from working properly under UNIX System V/386. These incompatibilities include:

- No support for quiet mode, no-echo mode, or auto-answer mode.
- Modem floods the bus with spurious interrupts.

When you are hooking up your modem, or any other device, make sure that serial wires connected to your computer are not left hanging. An unterminated line connected to your computer can considerably reduce system performance; always unplug a modem wire at the computer end instead of at the modem end.

Three-wire cables often used to connect terminals to the computer are not sufficient for connecting modems. For a modem cable on a 25-pin serial port, pins 2, 3, 7, 8, and 20 must be connected "straight-through" (meaning the pins are connected straight across: pin 2 to pin 2, pin 3 to pin 3, and so on). In addition, pins 4 and 5 must be connected straight through if RTS/CTS is used. If you are unsure as to what to use, a cable that connects all pins works correctly. Either a ribbon cable, or a straight-through cable will work. See [serial\(HW\)](#) for details of 9-pin connections.

To install your modem for dial-in or dial-out, follow these steps:

1. Make sure the UUCP package is installed. Use `custom(ADM)` to install it if necessary.
2. Make sure the serial port you have chosen for your modem is recognized at bootup (check `/usr/adm/messages` or use `hwconfig(C)`) and, if the modem is internal, make sure that the COM port does not conflict with any other device. Only serial devices attached to COM1 and COM2 are supported by SCO.
3. Make sure the modem and non-modem control ports are disabled by using the `disable(C)` command. For example, the following commands applies to a modem connected to COM1:

```
disable tty1a
disable tty1A
```



- If you are going to use the line for dial-out, the serial port must be owned by *uucp*. To make sure the line is owned by *uucp* enter these commands:

```
chown uucp /dev/tty m
```

```
chgrp uucp /dev/tty m
```

where *m* is the number of the corresponding line.

- Connect the modem to the machine using a straight-through cable (pins 2 and 3 are not crossed). The cable must have at least pins 2, 3, 7, 8, and 20 connected. (This is for a 25-pin serial port. See *serial(HW)* for 9-pin connections.)

Most standard COM ports use straight-through cables, but some hardware requires a null-modem cable (pins 2 and 3 are crossed). A standard COM port is known as DTE (Data Terminal Equipment); a port that needs a null-modem cable is known as DCE (Data Communications Equipment). Check your hardware documentation if you are unsure. If the COM board is a DCE, you need a null-modem cable.

- If you intend to use the modem for dial-in, you need to check the */etc/inittab* file and confirm it has an entry for your port similar to the following:

```
Se1A:2:off:/etc/getty -t60 tty1A m
```

*/etc/inittab* determines whether the port has a login prompt and defines the serial line characteristics. There should already be an entry for the line you are using; all you need to do is check the last field. This field is a number or letter (label) from the */etc/gettydefs* file, which associates a set of serial line characteristics (such as baud rate, parity, and number of stop bits). Use the label from that corresponds to the incoming baud rate you are using:

| Baud rate | Label  |
|-----------|--------|
| 1200      | 2      |
| 2400      | 3      |
| 9600      | m or 6 |

You can change */etc/inittab* with a text editor, but do not alter any fields other than the *gettydefs* label. For example, if you are using a 2400 baud modem on the first standard serial port, the line should be similar to the following:

```
Se1A:2:off:/etc/getty -t60 tty1A 3
```

The entry for a 9600 baud modem on COM2 should look like this:

```
Se2A:2:off:/etc/getty -t60 tty2A m
```

If you make any changes, use the following command to check the entries:

```
grep -i tty m /etc/inittab
```

8. Add the correct entries to the `/usr/lib/uucp/Devices` file. This file should have two entries for each serial port being used for a modem. One of the entries is used when you start a call using the modem (the Automatic Calling Unit (ACU) line), and the other line is used to connect directly with the modem to issue commands manually (the Direct line). For a Hayes-compatible modem operating at 2400 baud, using COM1, the entries in `/usr/lib/uucp/Devices` should be:

```
ACU tty1A - 1200-2400 /usr/lib/uucp/dialHA24 \T
Direct tty1a - 1200-2400 direct
```

Make sure that the entries do not have a number sign (#) in front of them. This is the syntax to show that the line is only a comment, and is to be ignored. There are many examples in the *Devices* file that are commented out with this character.

**WARNING** Be sure that any “Direct” entries in the *Devices* file appear *after* the corresponding ACU entries. If this is not so, the modem may not be reinitialized properly after an outgoing call.

9. Test the modem’s ability to dial correctly. (This process is outlined in Figure 16-1.) For example, the following command uses `tty1a`:

```
cu -l tty1a dir
```

You should see a message indicating that you are connected. If you see the message “cu: dir permission denied,” the user executing the `cu` command does not have write permission on the `/usr/lib/uucp/Devices` file.

If you do not see such a message, and there was no message to indicate that you connected correctly, check the following:

- ensure the “Direct” entry in the *Devices* file is correct
- make certain you entered the `cu` command correctly
- run the `hwconfig` utility and make sure your serial port is configured
- check the permissions on the port (see step 4)

**NOTE** The instructions that follow assume a Hayes-compatible command set and response codes. Other modems may use other conventions. Consult your modem documentation for further details.

10. If you see a message confirming your connection, enter the following on your keyboard:

```
AT
```

The screen should echo “OK”. If the modem is set to return result codes as numeric codes rather than text, you see “0”.

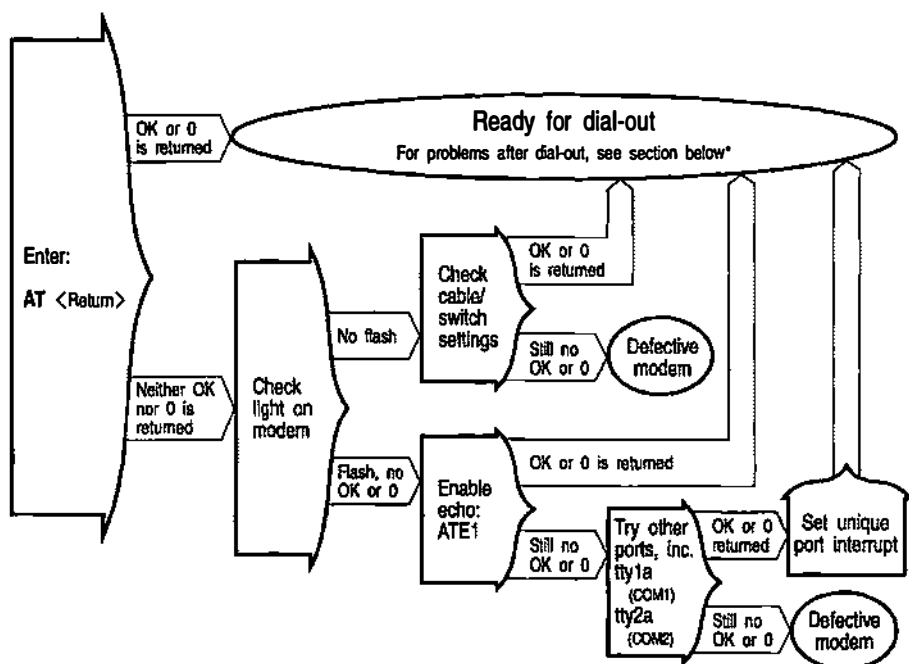


Figure 16-1 Testing your modem connection

11. If this does not occur, check that the "receive" light on the modem flashes when you press a key. This indicates the modem is receiving signals from the keyboard. If this light is not flashing, check your cable and modem switch settings. If the "receive" light flashes, but you still do not get an "OK" response from the modem, enable the modem's echo capability and responses to commands by entering:

**ATE1Q0**

12. If your terminal still does not display "OK" or "0", connect the modem to another port (COM1 or COM2). If the modem works with the new port, check that the device is using the correct interrupt vector. (Serial port COM1 uses interrupt 4; COM2 uses interrupt 3.)
13. If the terminal does not echo the "OK" message when it is connected to the new port, then double check your cable. You can also connect a terminal to the port (with a proper terminal cable) and verify that the port works. If the modem returns garbage, then try connecting at different speeds; some modems can be set to communicate at the DTE connect speed, while others communicate at a fixed speed. If the modem still does not respond properly, the modem may be defective.

14. If you plan to use the modem primarily for dial-in, skip to step 15. To test the modem for dial-out, enter:

**ATDT $\textit{phonenumber}$**  (for tone dial phone connection)

**ATDF $\textit{phonenumber}$**  (for pulse dial phone connection)

When you confirm that the modem can dial out, exit **cu** by entering:

~.

Then press (Return).

You are now ready to dial into another system. Use the following command to dial out:

**cu -x9 -lty1A  $\textit{phonenumber}$**

where  **$\textit{phonenumber}$**  is number of the remote system. The **-x9** option generates useful debugging information.

**NOTE** When including a phone number in the **cu** command to connect with a modem, you should not use the comma (,) from the Hayes command set to indicate a pause. Use hyphens instead. This also applies to *Systems* file entries.

15. If the line is also to be used for dial-in, you must follow some additional steps to configure the modem. Some modems have switches or software commands for setting the modem configuration. If your modem has such settings, configure it as instructed in your modem manual. The modem should be configured as follows:

- The modem must be set to Auto-answer. Your modem must support auto-answer; some modems do not have this setting.
- It should not answer when the DTR (Data Terminal Ready) line is low, and should disconnect from the current connection when DTR goes from high to low.
- The CD (Carrier Detect) line should follow the incoming carrier; it should go low when a carrier is present, high when a carrier is not present.
- Echo should be turned off; it should not echo characters sent to it.
- It should be set for quiet mode, sending no response codes.

16. Enable the port you are using for your modem with the **enable(C)** command:

**enable  $\textit{ttyname}$**

where  **$\textit{ttyname}$**  is the modem control port.

17. Dial this modem from another modem.

If you have any problems, refer to the section on troubleshooting modems in *Performance and Troubleshooting* in the *System Administrator's Guide*.

## Dialing into your computer

---

To allow dialing into your computer, you must enable a serial line that recognizes modem control signals with the `enable(C)` command.

To use the main serial adapter (COM1), enter:

```
disable tty1a
enable tty1A
```

For the alternate serial adapter (COM2), enter:

```
disable tty2a
enable tty2A
```

Note that `tty1A` and `tty1a` refer to the same (main) serial line, and `tty2A` and `tty2a` refer to the same (alternate) serial line. Do not enable the same line in both its modem and non-modem modes at the same time, as this causes an error.

## Adding passwords for dial-in lines

---

If desired, you can define special dial-in passwords on selected tty lines, requiring selected classes of users to input dial-in passwords. Logging information, including the last time of connection, can be stored for later use.

Specific dial-in lines that require passwords are defined in the file `/etc/dialups`. The format is one tty device name per line, for example:

```
/dev/tty1A
/dev/tty5C
```

The actual dialup passwords are kept in the file `/etc/d_passwd`. The password format is the same one used in `/etc/passwd`. The first field ("user name") in `/etc/d_passwd` is not a user name, but the name of a shell program (for example, `/bin/sh`) used in `/etc/passwd`. If the login shell of the user attempting to log in (on a tty line listed in `/etc/dialups`) is listed in `/etc/d_passwd`, then the user is prompted for the dial-in password stored in `/etc/d_passwd`.

Here is the syntax for creating a dial-in password:

```
passwd -m dialname
```

Change the password for dialup shell *dialname* (listed in `/etc/d_passwd`). If *dialname* begins with a slash (/) the entire shell name must match. Otherwise the password for every shell whose basename is *dialname* is changed. Only the super user can change a dialup shell password.

## Dialing out from your computer

The `cu(C)` and `uucp(C)` utilities call remote systems and transfer data on UNIX systems. The file `/usr/lib/uucp/Devices` (referred to as *Devices*) contains information used by these programs to determine the characteristics of a particular serial line.

The *Devices* file contains lines that specify the device for the line, the call-unit associated with the line, and the baud rate, which are to be used by UUCP. (Modem control devices should be used with lines connected to modems.)

### Using dialer programs

For dialing, both `cu` and UUCP use a common set of dialers, which can be standalone binaries (programs) like `/usr/lib/uucp/dialHA12`, or entries from the file `/usr/lib/uucp/Dialers`.

#### Dialers file entries

Table 16-3 lists dialer types that are available as *Dialers* entries.

**Table 16-3** Dialers file entries

| Dialer type | Modem or data switch                        |
|-------------|---------------------------------------------|
| direct      | direct line; no dialer                      |
| penril      | Penril modem                                |
| hayes1200   | Hayes Smartmodem 1200 (or compatible)       |
| hayes2400   | Hayes Smartmodem 2400 (or compatible)       |
| ventel      | Ventel 212+ modem                           |
| vadic       | Racal Vadic 3451 modem                      |
| vadic9600   | Vadic 9600VP                                |
| develcon    | Develcon network dataswitch                 |
| micom       | Micom network dataswitch                    |
| rixon       | Rixon Intelligent Modem                     |
| att4000     | AT&T Programmable 300/1200 Modem Model 4000 |
| att2212c    | AT&T DATAPHONE II 2212C Modem               |
| att2224     | AT&T DATAPHONE II 2224 Modem                |
| nls         | Network Listener Service                    |

The following is an example *Devices* entry that uses a *Dialers* file entry:

```
ACU tty1A - 2400 hayes2400
```

**NOTE** If you are experimenting with an unsupported modem, the “hayes2400” *Dialers* file entry is a good starting point; it can be used even with high-speed modems.

Although it is possible to create or modify a *Dialers* file yourself, the syntax can be difficult to follow. Consult the `dialers(F)` manual page for more information.

## Dialer binaries

Table 16-4 lists the dialer binaries provided with your distribution. They are supplied in binary and source form.

**Table 16-4** Dialer binaries

| Binary file | Modem                               |
|-------------|-------------------------------------|
| dialHA12    | Hayes Smartmodem 1200 or compatible |
| dialHA24    | Hayes Smartmodem 2400 or compatible |
| dialHA96V   | Hayes V-series Smartmodem 9600      |
| dialMUL     | Multitech MultiModem 224 EH or v.32 |
| dialVA3450  | Racal Vadic 3451 Modem              |
| dialT1500   | Trailblazer TB1500                  |
| dialTBIT    | Telebit Trailblazer Modem           |
| atdialHAY   | Hayes Ultra 96 Modem 2000US         |
| atdialMICC  | MICC 9610 Modem                     |
| atdialMT    | MULTITECH MT932EA Modem             |
| atdialT25   | Telebit T2500 Modem                 |
| atdialUSR   | US Robotics Courier V.32bis Modem   |
| atdialW96   | US Robotics World Port 9600 Modem   |

There are two types of dialer binaries: *dial* and *atdialer*. The *dial* binaries (example *dialHA24*) are separate executable programs that must be recompiled from source files if you wish to customize them. The *atdialer* dialers (example: *atdialHAY*) are actually a single executable (*/usr/lib/luucp/atdialer*) with a configuration file that can be changed with any text editor. The sections that follow explain how to create both types of dialers.

## Modem commands and setup strings

To create a new dialer, you need to understand how modem commands are used. You can enable or disable desired features by sending commands in a "setup string" to the modem. For example, the following setup string is used in the *dialHA24* dialer:

```
ATQ0E0T&D2&C1S0=0X4S2=043
```

You can change these strings to suit a different communications protocol or modem-specific commands by consulting the documentation for your modem. Though the setup commands may seem confusing because they are concatenated, there are two basic types of Hayes commands:

- modem commands (example: Q0, &D2)
- modem S-registers (example: S0=0)

Hayes-compatible command strings always begin with the AT (attention) command.

**NOTE** Consult your modem documentation for complete information on commands and options.

## Creating a new dial dialer

You can customize a *dial* program by editing the source file (example *dialHA24.c*) and changing modem commands embedded in the C code and then recompiling the source.

The source and a makefile for recompiling *dial* binaries are included in the directory */usr/lib/uucp*. If you have any other kind of modem, you can modify any of the source files and create your own dialer program. Remember that you must have the Development System installed to compile a *dial* program.

To make a new *dial* program, follow these steps:

1. Change directory to */usr/lib/uucp*.
2. Copy the dialer program you wish to modify and make the desired changes. For example, the *dialHA24.c* setup commands are located in this code segment:

```
#defineMDSETUP"ATQ0E0T&D2&C1S0=0X4S2=043\r"
```

There are several "defines" in the program, each containing a different string, such as MDHANGUP, the command string to send that tells the modem to hang up (ATQ0H). Change each define as appropriate.

**NOTE** The \r is not a modem command, but a string that indicates a carriage return; all modem commands in a dialer program require a carriage return as if they were being entered on the screen of your computer.

3. Edit the file *makefile* in the directory */usr/lib/uucp* and find the line that reads:

```
EXES= dialHA12 dialHA24 dialHA96V dialMUL dialTBIT dialT1500 dialVA3450
```

Add the name of the dialer program that you wish to use. When this is done, exit the file, saving the changes you made.



4. Next, enter the following command at your shell prompt:

**make**

When the **make** command is finished, you have a new dialer program. This can be used in the fifth field of an entry in the *Devices* field.

### *Creating a new atdialer dialer*

You can create a new *atdial* dialer without having the Development System installed. An *atdial* dialer is actually a link to the binary */usr/lib/uucp/atdialer* that calls a configuration file in the */etc/default* directory. The configuration file contains all the commands specific to that modem. For example, *atdialHAY* is linked to *atdialer* and the configuration file is in */etc/default/atdialHAY*.

**NOTE** *atdial* dialers are for high speed modems. Many different modems are available in the marketplace. They range in speed from 300 to 38400 bps. SCO does not support serial communication speeds greater than 9600 bps. In some cases, higher speeds are known to work. Some modems also have data compression while others do not.

**Table 16-5 atdialer configuration strings**

| String      | Meaning                | Example            |
|-------------|------------------------|--------------------|
| MDM_SETUP   | Setup string           | AT&FE0TL0&D2&C1&Q5 |
| MDM_OPTION  | Special option*        |                    |
| MDM_DIALCMD | Dial command           | ATD                |
| MDM_ESCAPE  | Escape to command mode | +++                |
| MDM_HANGUP  | Hangup                 | ATQ0H0             |
| MDM_RESET   | Reset                  | ATQ0Z0             |
| MDM_DIALIN  | Setup for dial-in      | ATS0=1Q1           |
| MDM_ATTN    | Attention              | AT                 |
| MDM_DSBLESC | Disable modem escape   | ATS2=128           |
| RTC_speed   | Result codes†          | CONNECT 9600       |

\* This feature allows you to set a special register when you put an X or an x at the end of the phone number in a *Systems* file entry. For example, you can send ATSP to do UUCP spoofing on a phone number with an "x" at the end. This feature is only valid when used in a *Systems* file entry; it cannot be used with a telephone number on the cu command line.

† Configuring the connect speed strings is an important aspect of creating a dialer. See "Editing the connect speed strings" (page 839) for more information.

## Modems that vary line speed

Some high speed modems can run the serial port at specified speeds, while the modem varies the line speed as needed. The modem must be prepared for this option by putting the appropriate commands in the `MDM_OPTION` entry then modifying `/usr/lib/uucp/Devices` and `/etc/inittab` to use the highest available speed. You must also edit the configuration file so that the strings containing the connect speed for all connections read "not used", except for the 9600 entry. See the next section for more information.

## Editing the connect speed strings

The `RTC_speed` strings in the `/etc/default/atdial*` configuration files allow the dialer to recognize connect messages from the modem and map them to appropriate speed(s). For example:

```
RTC_9600=CONNECT 9600
```

The dialer then recognizes `CONNECT 9600` as a 9600 bps connection. The connect messages for each speed must be unique or the first speed mapped to the connect message is used. For unused speeds, simply set the `RTC_speed` string to *not used*. For example:

```
RTC_300=not used
```

Normally, all but one `RTC_speed` string is set to *not used*. This is because most 9600 bps modems use "speed conversion". They communicate with the host serial port at 9600 bps even if they connect to the remote modem at a slower speed. The `RTC_speed` strings for these modems generally look like this:

```
RTC_9600=CONNECT
```

Any modem result message that begins with `CONNECT`, such as `CONNECT 9600` or even `CONNECT 2400`, is recognized as a 9600 bps connection and accepted. It is possible to refine this by specifying all valid connect messages. For example:

```
RTC_2400=CONNECT 2400
```

```
RTC_9600=CONNECT 9600
```

For a modem that performs speed conversion, this more accurate setup is only necessary if you need to guard against long UUCP transfers at 2400 bps. This is because the speed recognized by the dialer is matched against the speed range in the `Systems` file or the speed range on the `cu` command line. If it is out of range, an `EXECDIAL LOCAL FAILURE` will result. In the following command the speed range specified is 9600-9600:

```
cu -lty1A -s9600 5551212
```

Connections at 2400 bps using the above command will fail if separate *RTC\_speed* lines are set up. The following commands allow a 2400 bps connection in this case:

```
cu -lty1A -s2400 5551212
```

```
cu -lty1A -s2400-9600 5551212
```

### *RTS/CTS flow control*

Flow control is used between the computer and the modem to start and stop the flow of data when necessary (as when the modem's input buffer is full, or the computer needs to perform a disk write). There are two types of flow control: hardware (RTS/CTS) and software (XON/XOFF). RTS/CTS is the preferred and most efficient method.

The modem must also be configured for hardware flow control. For example, the *atdialHAY* dialer supports RTS/CTS flow control with the *&K3* command.

### *RTS/CTS on outgoing calls*

RTS/CTS flow control for outgoing calls is supported by the *atdialer*, *dialHA96V*, *dialMUL*, and *dialTBIT* binary dialers. All of these dialers recognize strings of the form *STTY=settings* in their */etc/default/dialer* files. For example, you can enable bidirectional hardware flow control for the *dialTBIT* dialer by creating a file */etc/default/dialTBIT* that contains the following line:

```
STTY= RTSFLOW CTSFLOW
```

### *RTS/CTS on incoming calls*

RTS/CTS flow control for incoming calls is configured by adding the *RTSFLOW* *CTSFLOW* flags to the appropriate */etc/gettydefs* entry specified in the */etc/inittab* entry for the modem tty port.

The *CTSFLOW* flag is often used by itself to prevent the modem buffer from overflowing when speed conversion is used for incoming calls. For example, assume a MultiModem v.32 is set for a fixed DTE speed of 9600 bps (the default). When a 2400 bps calls in, then the computer will send data to the modem at 9600 bps, which the modem can only send out at 2400 bps. The *CTSFLOW* flag in */etc/gettydefs* tells the serial driver to stop sending data when the modem drops CTS. This allows the modem to signal when it is busy and prevent data overruns.

***atdialer creation procedure***

To create a new *atdialer*, do the following:

1. Log in as *root*.
2. Change directories to */usr/lib/uucp*.
3. Enter the following command:

**ln atdialer atdialnew**

where *new* is the name of the dialer you wish to create.

4. Change directories to */etc/default*.
5. Copy an existing configuration file that you can modify. For example:

**cp atdialHAY atdialnew**

Be sure and use the same name you used in step 3.

6. Edit the configuration file as desired. Consult Table 16-4 for a list of file entries.

## Chapter 17

# Helpful Hints

---

This chapter answers questions concerning:

- broadcast addresses
- problems with the WD8003 card
- remote backups
- **sendmail** program
- user equivalence across systems

## Setting the broadcast address

---

**Q:** What should the host part of the broadcast address be?

**A:** The host part of the broadcast IP address should be all 0's with 4.1x or 4.2 BSD. In BSD 4.3, and versions derived from it such as SCO TCP/IP, the broadcast address has been corrected to use all 1's.

The INTERNET RFCs state that the host part of the IP address should be all 1's for broadcast purposes.

In releases prior to BSD 4.3, which were implemented before the standard was set, Berkeley implemented the broadcast address to be all 0's. Therefore, several Berkeley systems and TCP/IP implementations that are derived from BSD 4.1x and BSD 4.2 use all 0's for their broadcast address.

SCO supports both broadcast addresses; however, all hosts on a given network must have the same idea of what the broadcast address is. If you are running 4.3-derived TCP/IP, you still must use a broadcast address of 0's if you want to broadcast on the same network as BSD 4.2-derived implementations. Consequently, in BSD 4.3 and implementations derived from BSD

4.3, there is a broadcast option to the **ifconfig** command that allows you to specify which broadcast address you are actually going to use.

**ifconfig** is run from the */etc/tcp* script. See **ifconfig(ADMN)** in the Network Commands (ADMN) section of the *SCO TCP/IP* manual for more details.

### *Problem with WD8003 card*

---

**Q:** After installing SCO TCP/IP, I can ping **localhost** and send out packets, but I do not receive packets back from other machines. The WD8003 card is installed at interrupt vector 2, but it does not seem to be able to make the driver respond to its interrupts. If I ping another machine, I get 100% packet loss.

**A:** On some machines, the WD8003 card does not work at interrupt vector 2. This is because of a slightly non-standard bus specification on that machine.

You must reconfigure the device driver for the WD8003 card.

- Enter **netconfig**.
- Reconfigure the chain containing the card, and use interrupt vector 5. If there is a tape or a parallel port at that address, move it to vector 2.

Refer to the SCO system *Installation and Upgrade Guide* for instructions on reconfiguring device drivers.

You should test TCP/IP before adding any packages, such as NFS, on top of it. If you do add packages and need to reinstall the TCP/IP device driver, you will need to reinstall all the other packages as well.

### *Making remote backups*

---

**Q:** How do I make a remote backup over an SCO TCP/IP network?

**A:** The following commands demonstrate how to archive files across a network. *machinename* should be replaced with the remote hostname.

When making a backup, you should keep the following points in mind:

- The machines should have host equivalency. This means that the SCO TCP/IP */etc/hosts.equiv* file must contain the appropriate hostnames that are involved in the backup. It may also be necessary to have a *\$HOME/.rhosts* file depending on the UID of the user who is running the **rcmd**. See the SFF section of the SCO TCP/IP manual for more information on these files.

- If you are backing up whole filesystems, you should also consider that the **rcmd** command must be run as *root*, otherwise you do not have permission to read all the files.
- The **rcmd** command is executed from the current working directory. You may not want to include absolute pathnames in your backups. This means including a *./* or a *.* in the pathname specification, depending on what your current directory is.
- The examples shown cross mount points. That is, a filesystem that is mounted within the filesystem being backed up is also backed up.
- Although documented, **tar(C)** is not recommended for use as a backup mechanism because it does not save information about directories and special files. **tar** also cannot handle files whose names contain more than 100 characters. **cpio(C)** can handle filenames up to 256 characters in length.
- Neither **tar** nor **cpio** deals sensibly with sparse files (which cause a lot of data to be passed along the net).
- Backups on SCO TCP/IP for SCO UNIX System V/386 should only be made on SCO TCP/IP Release 1.1.1 or greater, as data corruption may occur. If you require a newer version, it is available as a free update.
- Backups should not be made on floppies, as they are limited in size and there is no easy way to load new disks as the backup continues. This may apply for small tape sizes as well.

## Backing up files or filesystems

---

If you want to perform a backup where the command is issued locally, but the files and the device are remote, you may use one of the following commands:

```
rcmd machinename tar cvf /dev/rct0 /usr
rcmd machinename "find /usr -cpio -print | cpio -oc > /dev/rct0"
```

If the files are local, and the device is remote, you may use one of the following commands:

```
tar cvf - /usr | rcmd machinename dd of=/dev/rct0
find /usr -cpio -print | cpio -oc | rcmd machinename dd of=/dev/rct0
```

If the files are remote, and the device is local, you may use one of the following commands:

```
rcmd machinename "tar cvf - /usr" > /dev/rct0
rcmd machinename "find /usr -cpio -print | cpio -oc" > /dev/rct0
```

## Restoring a backup

---

If you want to restore a backup where the command is issued locally, but the files and the device are remote, you may use one of the following commands. Note the changes to the **tar**, **cpio**, and **dd** commands.

```
rcmd machinename tar xvf /dev/rct0
rcmd machinename "cpio -idcv < /dev/rct0"
```

If the files are local, and the device is remote, you may use one of the following commands.

```
rcmd machinename dd if=/dev/rct0 | tar xvf -
rcmd machinename dd if=/dev/rct0 | cpio -idcv
```

If the files are remote, and the device is local, you may use one of the following commands:

```
dd if=/dev/rct0 | rcmd machinename tar xvf -
dd if=/dev/rct0 | rcmd machinename cpio -icdv
```

See the **tar**, **cpio**, and **rcmd(C)** manual pages for more information.

## Differences in sendmail implementations

---

**Q:** What are the differences between the */usr/lib/sendmail* provided with SCO UNIX System V/386 Release 3.2 and the */usr/lib/sendmail* provided with SCO TCP/IP?

**A:** The */usr/lib/sendmail* program provided with SCO UNIX System V/386 Release 3.2 is a front end to the MMDF mail system. It is provided for those third party programs that normally use the Berkeley **sendmail** program as their mail agent.

The */usr/lib/sendmail* provided in SCO TCP/IP, and installed with **mkdev sendmail-init**, is the traditional Berkeley **sendmail** program. This **sendmail**, if installed with **mkdev sendmail-init**, completely replaces the MMDF mail system. As such, they are mutually exclusive. However, it must be noted that it is not absolutely necessary to use **sendmail** under SCO TCP/IP, as SCO TCP/IP and MMDF are compatible. Berkeley **sendmail** is provided for those more familiar with a traditional mailer.



## Setting up user equivalence

---

**Q:** How do I set up user equivalence between two systems running SCO TCP/IP?

**A:** User equivalence is a state in which a particular user or group of users may access the accounts of another user or group of users, where this second group is usually on a different machine. This access is done without the use of any authentication, such as passwords.

For example, if user *alpha* on machine *m1* is equivalent to users *alpha* and *beta* on machine *m2*, then the following commands work without specifying any passwords from *alpha*'s account on *m1*. For example:

**rlogin m2**

User *alpha* logs into *m2*.

**rlogin m2 -l beta**

User *beta* logs into *m2*.

**rcmd m2 who**

User *alpha* executes the *who* command on machine *m2*.

**rcmd m2 -l beta who**

user *beta* executes the *who* command on machine *m2*.

Also, note the following:

**rcp filename m2:filename2**

This command requires user equivalence of user *alpha* on *m1* for user *alpha* on *m2*.

**rcp filename beta@m2:filename2**

This command requires user equivalence of user *alpha* on *m1* for user *beta* on *m2*.

There are two files that control this access. The first is *.rhosts* in the home directory of the affected user. The format of the *.rhosts* file is:

**machine username**

The username is optional.

The other file is */etc/hosts.equiv*. The format is identical to that of the *.rhosts* file, but usually only the *machine* portion is used.

For example, suppose user *alpha* on machine *m1* wants to allow user *alpha* on machine *m2* to access her account without the use of a password. User *alpha*, on *m1*, would create a file called *.rhosts* in her home directory with the line:

```
m2 alpha
```

In addition, if *alpha* wanted to allow user *delta* on *m2* and *gamma* on *m3* to access her account without a password, the *.rhosts* file in *alpha*'s home directory would read:

```
m2 alpha
m2 delta
m3 gamma
```

If *alpha* additionally wanted all users on machine *m4* to be able to access her account without a password, the *.rhosts* file would read:

```
m2 alpha
m2 delta
m3 gamma
m4
```

Suppose that the system administrator of machine *m1* wanted to allow all users on machine *m5* to access their own accounts on machine *m1*. This would be accomplished by adding the following line to the */etc/hosts.equiv* file on *m1*:

```
m5
```

Thus user *beta* on machine *m5* could access user *beta* on *m1* without the need for a password.

Note that */etc/hosts.equiv* does not work for the user *root*. If you wish to access the *root* user on *m1* from *m2* without a password, you must set up a *.rhosts* file in the */* directory on *m1*, with:

```
m2 root
```

Or, if you want a user other than *root* on *m2* to access *root* on machine *m1* without a password, use:

```
m2 user
```

Note that users on the machines with their own *.rhosts* file **must** have a password assigned. Also, if the system administrator has configured a */etc/hosts.equiv* file, the users on that system must have a password assigned to make use of the */etc/hosts.equiv* file. Finally, the *.rhosts* file in a particular user's home directory must be owned by that user.

# *Performance and Troubleshooting*

## **Preface** **857**

---

|                                   |     |
|-----------------------------------|-----|
| <b>Audience</b> .....             | 857 |
| <b>How to use this book</b> ..... | 857 |

### *Chapter 1*

## **Troubleshooting overview** **859**

---

|                                                   |     |
|---------------------------------------------------|-----|
| <b>Avoiding system problems</b> .....             | 859 |
| <b>Preparing to recover from problems</b> .....   | 861 |
| <b>Kernel error messages</b> .....                | 861 |
| <b>Recovering from a system crash</b> .....       | 863 |
| Recovering from a system panic .....              | 864 |
| Recovering after a power failure .....            | 865 |
| Verifying the security system after a crash ..... | 866 |

### *Chapter 2*

## **Performance management** **867**

---

|                                                             |     |
|-------------------------------------------------------------|-----|
| <b>General performance considerations</b> .....             | 867 |
| <b>Understanding your performance goals</b> .....           | 869 |
| <b>Excessive paging</b> .....                               | 870 |
| <b>Kernel tables are too small</b> .....                    | 871 |
| <b>Tuning disk I/O parameters</b> .....                     | 872 |
| <b>Optimizing serial I/O</b> .....                          | 875 |
| Increasing the value of TTHOG and NCLIST .....              | 876 |
| Changing the sio_fifocfl array .....                        | 877 |
| Relinking the kernel .....                                  | 879 |
| <b>Reducing memory dedicated to graphical clients</b> ..... | 880 |

### *Chapter 3*

## **Tools reference** **881**

---

|                                                           |     |
|-----------------------------------------------------------|-----|
| <b>configure(ADM) — reallocate kernel resources</b> ..... | 882 |
| Using the configure command line .....                    | 883 |
| Overriding configure limit warnings .....                 | 883 |
| <b>crash(ADM) — view memory</b> .....                     | 883 |

|                                                               |            |
|---------------------------------------------------------------|------------|
| Initializing crash .....                                      | 883        |
| crash strstat (statistics about STREAMS buffers) .....        | 884        |
| var (values of tunable parameters) .....                      | 886        |
| <b>fsck(ADM) — check and repair filesystems .....</b>         | <b>886</b> |
| <b>id tune(ADM) — reallocate kernel resources .....</b>       | <b>887</b> |
| <b>integrity(ADM) — check system file permissions .....</b>   | <b>887</b> |
| <b>ps(C) — check process activity .....</b>                   | <b>888</b> |
| <b>sar(ADM) — system activity reporter .....</b>              | <b>889</b> |
| How sar works .....                                           | 889        |
| Running sar .....                                             | 890        |
| sar -a — file access operations .....                         | 891        |
| sar -b — buffer activity .....                                | 891        |
| sar -c — system call activity .....                           | 892        |
| sar -d — block device activity .....                          | 893        |
| sar -m — interprocess communication .....                     | 894        |
| sar -n — name cache activity .....                            | 895        |
| sar -p — paging activity .....                                | 895        |
| sar -q — process throughput .....                             | 896        |
| sar -r — page and swap usage .....                            | 897        |
| sar -u — CPU utilization .....                                | 897        |
| sar -v — system tables .....                                  | 898        |
| sar -w — swapping activity .....                              | 899        |
| sar -y — terminal I/O .....                                   | 900        |
| <b>swap(ADM) — check and add swap space .....</b>             | <b>901</b> |
| <b>timex(ADM) — examine system activity per command .....</b> | <b>901</b> |
| <b>tunesh(ADM) — automatic configuration tool .....</b>       | <b>902</b> |
| <b>XENIX system tools .....</b>                               | <b>903</b> |

## Chapter 4

# ***Kernel parameter reference*** **905**

---



---

|                                                         |            |
|---------------------------------------------------------|------------|
| <b>Disks and buffers .....</b>                          | <b>908</b> |
| <b>Character buffers .....</b>                          | <b>911</b> |
| <b>Files, inodes, and filesystems .....</b>             | <b>912</b> |
| <b>Processes, memory management, and swapping .....</b> | <b>915</b> |
| <b>Clock .....</b>                                      | <b>918</b> |
| <b>MultiScreens .....</b>                               | <b>918</b> |
| <b>Message queues .....</b>                             | <b>919</b> |
| <b>Semaphores .....</b>                                 | <b>920</b> |
| <b>Shared data .....</b>                                | <b>921</b> |

|                                             |     |
|---------------------------------------------|-----|
| <b>System name</b> .....                    | 922 |
| <b>STREAMS data</b> .....                   | 922 |
| <b>Event queues and devices</b> .....       | 925 |
| <b>Hardware dependent parameters</b> .....  | 926 |
| <b>Security</b> .....                       | 927 |
| <b>Asynchronous I/O</b> .....               | 928 |
| <b>Boot load extension parameters</b> ..... | 929 |
| <b>Multiprocessing parameters</b> .....     | 929 |
| <b>Network-specific parameters</b> .....    | 930 |
| TCP/IP parameters .....                     | 930 |
| LAN Manager parameters .....                | 930 |

## Chapter 5

# Troubleshooting reference

931

|                                                   |     |
|---------------------------------------------------|-----|
| <b>Boot troubleshooting</b> .....                 | 931 |
| Restoring missing or corrupted system files ..... | 932 |
| /boot not found .....                             | 932 |
| unix not found .....                              | 933 |
| Cannot load floating point emulator .....         | 934 |
| System hangs at login prompt .....                | 935 |
| System cannot enter multiuser mode .....          | 936 |
| execvp of /bin/sulogin failed .....               | 937 |
| /etc/bcheckrc not found .....                     | 937 |
| System hangs at boot time .....                   | 938 |
| <b>Console troubleshooting</b> .....              | 939 |
| Console keyboard locks up .....                   | 939 |
| Preventing console keyboard lockup .....          | 940 |
| Cannot log into console .....                     | 941 |
| <b>cron, at, and batch troubleshooting</b> .....  | 941 |
| cron daemon is not running .....                  | 941 |
| at command fails: cannot change mode of job ..... | 942 |
| <b>Disk problems</b> .....                        | 942 |
| Mapping a bad track .....                         | 942 |
| Disk load balancing .....                         | 943 |
| <b>Filesystem troubleshooting</b> .....           | 944 |
| Out of inodes on filesystem .....                 | 944 |
| Out of space on filesystem .....                  | 945 |
| Checking free space on filesystems .....          | 947 |
| Reducing disk fragmentation .....                 | 947 |

|                                                                           |     |
|---------------------------------------------------------------------------|-----|
| Restoring a corrupted root filesystem .....                               | 948 |
| Restoring your root filesystem from backups .....                         | 949 |
| Restoring a root filesystem with an Emergency Boot<br>Floppy Set .....    | 949 |
| Restoring a root filesystem without an Emergency Boot<br>Floppy Set ..... | 950 |
| Repairing a filesystem when fsck stops at size check .....                | 951 |
| Filesystem mount failed .....                                             | 954 |
| Unable to remove files .....                                              | 955 |
| <b>Login problems</b> .....                                               | 955 |
| Cannot log in after entering multiuser mode .....                         | 955 |
| Users cannot log into the system .....                                    | 956 |
| Login incorrect .....                                                     | 956 |
| useshell: file access control database inconsistency .....                | 956 |
| Unlocking a locked user account .....                                     | 956 |
| Replacing a forgotten user password .....                                 | 956 |
| <b>Mail troubleshooting</b> .....                                         | 957 |
| Failed mail error .....                                                   | 957 |
| Mail does not work, no returned mail .....                                | 958 |
| Mail command hangs .....                                                  | 958 |
| <b>Modem problems</b> .....                                               | 959 |
| <b>Modem — errors when dialing out</b> .....                              | 959 |
| No OK message (Hayes-compatible modems) .....                             | 959 |
| Modem dials, but does not connect .....                                   | 960 |
| Connect failed: NO DEVICES AVAILABLE .....                                | 960 |
| Modem answers, but terminal displays garbage .....                        | 961 |
| DEVICE LOCKED .....                                                       | 961 |
| Modem does not hang up .....                                              | 961 |
| Double echo .....                                                         | 962 |
| <b>Modem — problems dialing in</b> .....                                  | 962 |
| Modem does not answer the phone .....                                     | 962 |
| Modem answers, but hangs up .....                                         | 963 |
| Garbage or loose cable .....                                              | 963 |
| Modem answers, but no login prompt .....                                  | 964 |
| Screen displays a series of login prompts .....                           | 964 |
| System displays meaningless characters .....                              | 964 |
| <b>Mouse (serial) troubleshooting</b> .....                               | 965 |
| usemouse: no mouse available(5) .....                                     | 965 |
| usemouse: can't open slave (errno 13) .....                               | 965 |
| <b>Mouse (X server) troubleshooting</b> .....                             | 966 |
| <b>No more processes error message</b> .....                              | 966 |
| Runaway processes .....                                                   | 967 |

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| <b>Print system troubleshooting</b>                                      | 967 |
| lpsched print scheduler is not running                                   | 967 |
| Printer does not print                                                   | 968 |
| Cannot redirect output to printer                                        | 969 |
| Port does not respond                                                    | 970 |
| Printer output is illegible                                              | 971 |
| Printer output spacing is wrong                                          | 972 |
| Parallel printer is slow                                                 | 972 |
| Setting up polling                                                       | 972 |
| Changing the MODE_SELECT parameter                                       | 974 |
| Printer reports UUCP errors                                              | 974 |
| <b>Runaway processes, stopping</b>                                       | 974 |
| <b>Security system troubleshooting</b>                                   | 975 |
| Using the override terminal                                              | 975 |
| Account is disabled – see Account Administrator                          | 976 |
| Account is disabled but console login is allowed                         | 977 |
| Terminal is disabled but root login is allowed                           | 977 |
| Audit: filesystem is getting full                                        | 977 |
| Authentication database contains an inconsistency                        | 977 |
| Can't rewrite terminal control entry for tty                             | 978 |
| Authentication error; see Account Administrator                          | 978 |
| Cannot obtain database information on this terminal                      | 978 |
| Login incorrect                                                          | 978 |
| login: resource Authorization name file could not be allocated due to:   | 978 |
| cannot open;                                                             | 978 |
| Terminal is disabled – see Account Administrator                         | 979 |
| useshell: File Control database inconsistency                            | 979 |
| useshell: Script path /usr/lib/mkuser/mkuser.init may be compromised ... | 979 |
| You do not have authorization to run ...                                 | 979 |
| <b>Tape drive problems</b>                                               | 979 |
| Bad octal digit                                                          | 980 |
| Tape not recognized at bootup                                            | 980 |
| Cartridge tape                                                           | 980 |
| Irwin and QIC-40/80                                                      | 980 |
| SCSI tape                                                                | 981 |
| Tape commands hang                                                       | 981 |
| Cannot open /dev/rct0 error message                                      | 981 |
| <b>Terminal troubleshooting</b>                                          | 982 |
| Restoring a non-functional terminal                                      | 982 |
| Fixing a hung terminal                                                   | 983 |
| Fixing a scrambled terminal display                                      | 984 |



|                                                             |     |
|-------------------------------------------------------------|-----|
| Unlocking a locked terminal .....                           | 985 |
| Restoring a non-echoing terminal .....                      | 985 |
| <b>UUCP troubleshooting</b> .....                           | 985 |
| Check for faulty ACU/Modem .....                            | 986 |
| Errors when testing the connection with cu .....            | 987 |
| Connect failed: CANNOT ACCESS DEVICE .....                  | 987 |
| Connect failed: SYSTEM NOT IN Systems FILE .....            | 987 |
| Connect failed, NO DEVICES AVAILABLE .....                  | 988 |
| Connected, but no login prompt .....                        | 988 |
| Connected, but screen displays garbage .....                | 988 |
| UUCP failed messages .....                                  | 988 |
| Checking the status of a uucp request .....                 | 989 |
| Debugging UUCP communications .....                         | 990 |
| Alarms in UUCP audit output, data is not transferring ..... | 991 |
| UUCP log and status file messages .....                     | 991 |
| DEVICE LOCKED .....                                         | 992 |
| LOGIN FAILED .....                                          | 992 |
| NO DEVICES AVAILABLE .....                                  | 992 |
| REMOTE DOES NOT KNOW ME .....                               | 993 |
| REMOTE HAS a LCK FILE FOR ME .....                          | 993 |
| SYSTEM NOT IN Systems FILE .....                            | 993 |
| RETRY TIME NOT REACHED .....                                | 993 |
| CANNOT ACCESS FILE .....                                    | 993 |
| WRONG TIME TO CALL .....                                    | 994 |
| Checking UUCP files and permissions settings .....          | 994 |
| Verifying that sitename is unique .....                     | 994 |
| UUCP truncates system names to seven characters .....       | 994 |
| UUCP is abnormally slow .....                               | 995 |
| uucp works, but uux does not .....                          | 996 |
| UUCP troubleshooting utilities .....                        | 996 |
| <b>Index</b> .....                                          | 997 |

## *Preface*

---

This book provides information to help you get the highest performance possible from your machine and recover from problems that result from a power failure, system failure, or human error. The information in this book applies to the whole SCO® Open Desktop®/SCO® Open Server™ system.

## *Audience*

---

This book is written for system administrators. Performance management and troubleshooting requires a certain sophistication about the internal workings of your system; while we have attempted to provide the information needed for these tasks, users who are attempting to enhance the performance of their systems and troubleshoot error conditions may need to consult other sources to get a fuller understanding of how the system works.

## *How to use this book*

---

The book is designed to be used as follows:

- Use the index when you are seeking information about a specific topic; the index includes entries for all error messages that are discussed in the book and other detailed entries to enable you to find the information you need quickly.
- Peruse sections of this book leisurely before you have problems; understanding this material may help you avoid serious problems.

This book is arranged as follows:

**Troubleshooting overview**

provides general information about avoiding system problems and being prepared to handle problems quickly. It also gives detailed instructions for recovering from a system panic or power outage.

**Performance management**

discusses general steps you can take to improve system performance and discusses the general approach you should take to resolve performance problems. Many different situations can cause a degradation in performance; to solve the problem effectively, you must first analyze it to determine what the underlying cause is. If you determine that your system needs to be reconfigured to allocate system resources differently, this chapter also includes information about the adjustments that are most commonly required.

**Tools reference**

alphabetically arranged material about system tools that help you analyze and correct system problems

**Kernel parameters reference**

comprehensive listing of tunable parameters whose value can be modified to reallocate kernel resources appropriately. These parameters are discussed in the groupings and order in which they appear when you run the `configure(ADM)` program.

**Troubleshooting reference**

reference sections about various topics that can cause system problems. These sections are arranged alphabetically.

## Chapter 1

# Troubleshooting overview

---

Troubleshooting is the general term for studying and resolving system problems. Most commonly, troubleshooting concerns correcting errors that led to a system crash or processes failing. It can also involve resolving performance problems (some of which may eventually cause the system to crash or processes to fail). The following chapter gives an overview of how to study and resolve performance problems; this chapter discusses methods for studying other system problems.

Chapter 5 contains miscellaneous notes about problems that may arise, their causes, and how to correct them. Additional troubleshooting information is located in appropriate sections of the documentation set, including:

- *Installation and Upgrade Guide* contains information about solving problems that arise when installing the SCO Open Desktop/SCO Open Server software.
- *Administering Networking Services* contains networking troubleshooting information.

The *Graphical Environment Administrator's Guide* includes detailed instructions for using the graphical environment correctly.

## Avoiding system problems

---

Some system problems are unavoidable, but there are a number of actions you can take that will allow you to catch errors before they cause serious damage and to be able to recover from system problems quickly. These include:

- Always allow `fsck(ADM)` to run on all "dirty" file systems before they are mounted, especially after a system crash. `fsck` repairs a number of small filesystem errors before they cause serious damage.

- Monitor system performance regularly. This enables you to reallocate resources before they are exhausted and cause processes to fail or the system to crash.
- Monitor system error logs regularly so you can resolve problems before they cause serious system or process failure. The `/usr/adm/messages` file contains all kernel and hardware messages written to the console since the file was purged. You can also use the XENIX `dmesg(ADM)` command to view all console error messages that appeared since the system was last booted. The `strace(ADM)` command displays all STREAMS error messages that have been posted, and the `logger(TC)` command displays TCP/IP error messages.
- Keep a system log, and note all information that could be of interest in troubleshooting the system. Record all system errors that occur, all corrective and preventative actions taken, every time you change the hardware or software configuration, and so forth. For each event, log what happened, when it happened, who was involved, and any information you have about why it happened. Study this log periodically to spot trends that might indicate a developing problem.
- Always save a memory dump after a system crash. When you reboot the system, you may see the following message:

There may be a system dump memory image in the swap device.  
Do you want to save a copy of it? (y/n)

You can save this dump and study it with the `crash(ADM)` command. See “`crash(ADM)` — view memory” (page 883). We recommend that you archive dumps from previous crashes; this can provide a valuable history of developing problems. Old memory dumps can be archived to tape if necessary if you are short of disk space, but they must be written to disk before you can run `crash` on them.

The system writes the memory dump to `dumpdev`, which is normally the same as `swapdev`. For the memory dump to be saved correctly, `dumpdev` must contain at least as much swap space as you have physical memory.

- Depending on how your system is configured, the security system may enforce permissions on system files, so take care when changing permissions on system programs and directories. See “`integrity(ADM)`” (page 887) for more information.

## Preparing to recover from problems

---

In addition to taking steps to avoid problems, you can take steps that will enable you to recover from system problems more quickly and smoothly:

- Back up the system regularly. Consider running full system backups every week or two. Because incremental backups are usually faster than full backups, it is tempting to run full backups only rarely. However, if you need to restore files, you may need to restore the most recent full backup plus all incrementals run since then, which could be a long, tedious process.
- Make the Emergency Boot Floppy Set as discussed in the *Installation and Upgrade Guide*. If the *root* filesystem or a critical booting file are corrupted by a system crash, the Emergency Boot Floppy Set can be used to recover the system. Otherwise, you may need to reinstall the software before you will be able to boot the system.
- Set up the system so `fsck(ADM)` fixes any filesystem problems whenever the system is rebooted. See “`fsck(ADM)` — check and repair filesystems” (page 886) for more information.
- Understand the purpose of the “override” terminal as discussed in “Using the override terminal” (page 975). The override terminal is used to correct errors that prevent access to other terminals. By default, the console multiscree (`/dev/tty01`) is the override terminal.

## Kernel error messages

---

Kernel error messages report driver errors and errors in other parts of the kernel such as the process scheduling subsystem and the file subsystem. These error messages appear on the console, and are usually recorded in the `/usr/adm/messages` file. Monitoring these messages regularly is an important step in preventing serious system problems; studying these messages after a system problem develops is an important part of troubleshooting the system.

Kernel error messages usually have the format:

*class*: [*driver*: ] [*routine*] *message*

*class* is usually one of the following:

- CONFIG** Identifies a problem that can be corrected by modifying the system configuration. Often, such messages indicate that a kernel resource under control of a tunable parameter has been exceeded. The *message* tells what the resource is and identifies the tunable parameter that controls its size. See “Kernel tables are too small” (page 871) for more information.

**NOTICE** An unexpected situation has occurred. Typically, **NOTICE** errors are less serious than **WARNING** errors in terms of system stability, although they may indicate an error that has serious ramifications for executing applications.

**WARNING** An error condition has occurred.

**PANIC** The kernel is corrupted and the system shuts down to prevent additional problems.

*driver* names the device having problems and is usually the same as the internal name defined in column 1 of *mdevice(F)*. The actual peripheral is usually identified by a pair of numbers of the form *major/minor*. This identifies the device number of the peripheral where the error occurred. The major and minor numbers also appear in the `ls -l` listing on a */dev* file, in the column just before the date where the byte count appears on regular files. For example, the following output shows that 2 is the major number and 52 is the minor number of */dev/fd0*:

```
brw-rw-rw- 7 bin bin 2, 52 Jun 11 1990 /dev/fd0
```

The *routine* element indicates the subsystem that detected the condition; these portions of the error message are included mostly to help support staff in tracking difficult system problems.

The *message* explains what the error condition is. In most cases, this should be enough to identify the problem. The following examples illustrate kernel error messages:

**WARNING:** floppy: Read error on dev 2/52, block=736 cmd=0x03 status=0x01

This tells you that there was a read error on the floppy disk. Its severity depends on the importance of the data being read at the time. Usually this indicates that the floppy has a bad spot, although sometimes it indicates filesystem or disk corruption on the device that holds the data being read. When you get a read or write error on a device, you should verify that you have good data on the device; it may be necessary to repeat the copy operation.

**CONFIG:** *routine* - Out of clists (NCLIST = 32 exceeded; 2 failures so far)

This message tells you that you have exceeded a system resource (*clist*, in this case) that is controlled by a tunable parameter (*NCLIST*). It also tells you the current value of the tunable, and in this case, the number of overflow conditions that have occurred. You can use the index of this book to locate information about the specific tunable parameter.

NOTICE: *routine* - No space on *fstype* dev *xname* (*major/minor*)

This message tells you that a filesystem has run out of space. *fstype* is the filesystem type (for example, EAFS, S51K, XENIX, or DOS). The *major/minor* information identifies the disk partition on which the filesystem is located, so by extension indicates the filesystem that has run out of space. This error message is discussed more in "Out of space on filesystem" (page 945).

PANIC: *s5iread* - Allocated *fstype* inode *num* on dev *xname* (*major/minor*) in free ilist

PANIC: *s5iupdat* - *filtyp* inode *num* address *num* > 2<sup>24</sup> on *fstype* dev *xname* (*major/minor*)

Error messages such as these two PANIC messages are not as easy to understand. *xname* is usually the same as the *driver* field discussed above. These two messages come from the file subsystem, which uses inodes to access a file. Normal conditions usually do not cause such problems, so messages such as these may indicate a bad memory board or some other hardware problem.

## Recovering from a system crash

---

A system crash is any time the system goes down without closing files and doing other cleanup operations. Two types of system crashes occur:

- panic      The system "panics" when it encounters a hardware problem or kernel inconsistency that is so severe that the system cannot continue functioning.
- powerfail   If the AC power to the system fails, even briefly, the system crashes.

When your system goes down because of a panic or power failure, the shutdown is considered "abnormal." A third type of abnormal shutdown occurs when the operator must crash the system, usually because a program goes into a loop or the system hangs. When the system is hung because of an application program error, you should crash it as soon as possible; the longer the loop goes on, the greater the possibility of file damage. Recovering from an operator-induced crash is similar to recovering from a system panic.

When the system is shut down normally, the `shutdown(ADM)` program stops all daemons, kills the active processes, unmounts any mounted filesystems, and runs the `sync` command, and tells `init` to bring the system down to the appropriate state (either single-user or "safe to power off"), or to reboot.



If the system goes down before this shutdown procedure completes, the following may occur:

- Filesystems may be inconsistent or corrupted, resulting in lost data.
- Ongoing work by users and other data may be lost because the buffer cache was not flushed to disk.

UNIX System V synchronizes the disk with memory (`sync`'s) frequently, so the amount of data lost due to system shutdown without the `sync` command should be minimal. However, because the filesystems were not unmounted properly before the system shut down, filesystem corruption can occur. If the *root* filesystem is corrupted, the system may not function properly.

### *Recovering from a system panic*

---

The system “panics” when it encounters a hardware problem or kernel inconsistency that is so severe the system cannot continue to function. When this happens, the system usually displays a message on the console, and all system activity stops.

The system panic messages begin with the word “PANIC,” followed by a diagnostic message. Sometimes this diagnostic explains what caused the panic, but sometimes it only indicates that the kernel was corrupted and the actual cause of the corruption was a few instructions back. The system can also panic without displaying a PANIC message. When this happens, the system simply refuses to process any input from the system console and all other terminals.

To recover from a system panic:

1. Copy the full PANIC message (including CPU registers), if any, from the console screen to your system log book.
2. Power-cycle the machine and press `<Return>` at the “Boot:” prompt to reboot the system.
3. At the prompt to check the *root* filesystem, answer “y”.
4. Bring the system up in single-user mode and study the problem that caused the system to panic. If possible, fix the problem.
5. When the system prompts you, respond “yes” to save a copy of memory at the time of the PANIC. The system writes this copy to the *dumpdev* device, which is usually the same as the *swap* device, so the data will be overwritten as soon as any paging occurs on the system. If you respond “yes” to save a copy of memory, it is written to either floppy or tape. You can later copy this data to disk and use the `crash(ADM)` command to study the information. See “`crash(ADM)` — view memory” (page 883).

6. Run `fsck(ADM)` on those filesystems that were mounted when the system panicked. This happens automatically for all filesystems that are marked dirty when the system is brought up multiuser, but by running `fsck` manually, you can control the response to problems that are found; see “`fsck(ADM)` — check and repair filesystems” (page 886) for information. Any filesystems that are not checked automatically can be checked by issuing the `fsck` command when the system is in single-user state. For example, use `fsck /dev/u` to check the `/u` filesystem.
7. Verify the integrity of the security system. See “Verifying the security system after a crash” (page 866) for more information.

In rare cases, the operating system becomes corrupted and must be restored or reinstalled. Refer to “Restoring a corrupted root filesystem” (page 948) and “Boot troubleshooting” (page 931) for more information. In most cases, simply rebooting the system solves the problem. However, if your system consistently shuts down with the same PANIC message, you should fix the problem that is causing the system to panic.

## Recovering after a power failure

---

When the power fails, turn the machine off. This minimizes potential damage to your system if the power fluctuates.

Once the power comes back on, restore your system using the following steps:

1. Turn on the computer and press `(Return)` at the “Boot:” prompt.
2. If the `root` filesystem is corrupted, the system prompts you whether to check the `/dev/root` filesystem; enter “y” to invoke `fsck(ADM)`. The system may display the following message:

```
FREE INODE COUNT WRONG IN SUPERBLK
FIX?
```

This message is routine when the system is not shut down properly. If the system is not set up to automatically try to fix all `root` filesystem corruption (see “`fsck(ADM)` — check and repair filesystems” (page 886) for more information), enter “y” and `fsck` fixes the problem.

3. If any non-root filesystems are corrupted when the system enters multi-user mode, the system prompts you whether to clean the filesystem; enter “y”. If the system is not set up to automatically fix filesystem corruption (see “`fsck(ADM)` — check and repair filesystems” (page 886) for more information), you will be prompted each time `fsck` finds an error as to whether or not you want the corruption fixed.

## Verifying the security system after a crash

---

A system crash may damage the Authentication database of files that control system security. Several programs are used to maintain the Authentication database, the system area of the filesystem, and the filesystem as a whole. The basic rule is to work from the most basic components of the filesystem outward. Otherwise, corrections made at the higher levels may be undone by programs fixing the lower levels.

After a system crash or abnormality, run the following programs in this order while the system is in single user (system maintenance) mode.

1. Run **fsck(ADM)** to check filesystem integrity (automatic when rebooting to multiuser state).
2. Run **tcback(ADM)** to check for the absence and integrity of critical security files (automatic when rebooting to multiuser state). **tcback** calls **authck(ADM)** to attempt to repair any file corruption it detects. Administrators can also run **authck** from the command line or with the Accounts ⇨ Check ⇨ Databases selection from **sysadmsh**. When the **authck** command is run from the command line, you are given the option of allowing **authck** to repair any inconsistencies it finds. If corruption is found when you check the security files with **sysadmsh**, you can execute **authck** from the command line or restore files if necessary.
3. Make the Accounts ⇨ Check ⇨ Password selection from **sysadmsh** to check the consistency of the */etc/passwd* and */etc/group* files. This selection does the same checking that is done when a new user is created. If any error messages are generated, they must be acted upon.
4. Check system file permissions with **integrity(ADM)**. See “**integrity(ADM)**” (page 887) for more information.
5. Run **fixmog(ADM)** to fix file permissions inconsistencies found by **integrity**. **fixmog** changes the owner, group and access permissions of files to match the information in the File Control database. Always use the **-i** (interactive) option to ensure that you can confirm any changes before they are made.

See the chapter on starting and stopping the system in *Administering Operating System Services* for more information about the security-related aspects of the startup process.

## Chapter 2

# Performance management

---

Performance management involves budgeting system resources so they are most appropriately allocated for the way you use your computer. The **tunesh(ADM)** utility configures adequate resources for an “average” system usage pattern, with enough resources of all kinds allocated to enable most applications to run. This means that most users get satisfactory performance without taking special steps to enhance system performance. It also means that most users who want to study their system’s performance and reconfigure it accordingly can improve their performance, sometimes significantly.

UNIX System V includes tools for studying how system resources are being used on your system. The most important of these tools is **sar(ADM)** which reports on usage statistics. See “**sar(ADM)**” (page 889) for general information about setting up and studying **sar**; details about how to use these statistics are in the appropriate sections of this book. Information about performance management tools is arranged alphabetically by tool in Chapter 3.

## General performance considerations

---

The way you use and administer your system can impact system performance. Some factors to consider are:

### Run large, resource-intensive jobs during off-hours

Jobs such as backups, source-code builds, and database archive and reorganization operations can be run when most users are off the system.

- Use the **sar -u** report to determine hours when the system is normally quiet.
- Examine the files in *crontabs* and *atjobs* under the */usr/spool/cron* directory to see if tasks being run during prime time could be run at times when the system is idle.

- Tasks that transfer large amounts of data to or from a tape or networking device should be scheduled for off-hours whenever possible, because in many cases they degrade overall system performance when they run. Examples include backing up filesystems to tape and receiving the USENET news database.
- If the system seems "bogged down," run the `ps(C)` command to identify resource-intensive processes that are running. See if these processes could be run during quiet times, perhaps using `cron(C)` or `at(C)`.

### **Reorganize filesystems periodically to reduce disk fragmentation**

Fragmentation is a bigger problem when filesystems become more than 75% full. If possible, configure the disk partitions and arrange directories on disk so that filesystems maintain 15-25% of their blocks free at all times. Any time a filesystem has more than 90% of its blocks full, expect to suffer serious performance degradation. See "Reducing disk fragmentation" (page 947) for more information.

### **Repair hardware promptly**

Faulty hardware can seriously degrade system performance. Monitoring the `/usr/adm/messages` file for signs of faulty hardware can help you identify hardware problems that are developing. You can also monitor the `sar -y` report; when the "mdmin/s" field shows values greater than 0, it may indicate that modems, ports, or terminals need repair.

### **Keep user PATH efficient**

The `PATH` environment variable lists the pathnames of all directories to be searched each time a command is executed. Before displaying "not found," the system must search every directory in `PATH`. These searches require both processor and disk time. All the shells that run on SCO Open Desktop/SCO Open Server and UNIX System V support path caching which reduces the impact of inefficient `PATH` definitions, but individual users may still improve response time by keeping the `PATH` line as short as possible, with the most commonly-accessed directories (usually `/bin` and `/usr/bin`) at the beginning of the line. Avoid including large directories other than `/bin` and `/usr/bin` in the `PATH` line; if large directories must be included, put them at the end of the line.

The default `PATH` is defined in the `/etc/default/login` file; users may then redefine the default `PATH` in their `.login`, `.cshrc`, or `.profile` files. (See `environ(M)` for more information on environment variables.)

## Understanding your performance goals

The first step in performance management is to study the performance of your system and determine where potential or real performance bottlenecks are. These are the areas where you should concentrate your efforts.

Table 2-1 summarizes the symptoms and possible solutions for these performance goals.

**Table 2-1 Performance goals summary**

| Goals                    | Symptoms                                                                                                                                    | Solutions                                                                                                                                                                                                                                                                        |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Excessive paging         | <code>sar -q: "%swpocc"&gt;20</code><br><code>sar -w: "swpin/s" &gt; 1.00</code><br><code>swap -l: &lt;50% of swap free during peaks</code> | Reduce size of kernel (tune table sizes and number of buffers)<br>Increase swap space ( <code>swap -a</code> )<br>Add memory<br>Run fewer graphical clients<br>Move users to another machine                                                                                     |
| Kernel tables too small  | Console error messages<br><code>sar -v</code> shows overflows                                                                               | Tune tables to acceptable sizes                                                                                                                                                                                                                                                  |
| Improve disk performance | <code>sar -u: "%wio" &gt; 10%</code>                                                                                                        | Use ACER filesystems (default)<br>Balance filesystems across disks and controllers or host adapters.<br>Keep filesystems < 90% full<br>Reorganize directories<br>Keep directories small<br>Tune I/O parameters<br>Add more disks<br>Use faster disks, controllers, host adapters |
|                          | <code>sar -b: "%rcache" &lt; 90</code>                                                                                                      | Increase buffers (NBUF)                                                                                                                                                                                                                                                          |
|                          | <code>sar -b: "%wcache" &lt; 65</code>                                                                                                      | Tune name cache parameters                                                                                                                                                                                                                                                       |

The following sections discuss these goals in detail.

# Excessive paging

---

User processes execute in the portion of memory that is not occupied by the kernel, plus the swap area(s) configured on your disk(s). If the amount of main memory available for user processes is inadequate, pages of processes are written out to the *swap* device on disk. This transfer of pages between memory and the disk is costly in both disk and CPU overhead. To see whether this is causing degraded performance on your system, look at the following:

- page faults (vflt/s) greater than 100 on the **sar -p** report
- free memory (freemem) less than 100 pages on the **sar -r** report
- pages read from the swap device (swpin/s) greater than 1 and idle time (%idle) greater than 0 on the **sar -wu** report
- CPU utilization on the **sar -u** report. As a general guideline, if the percent of time the CPU was occupied by system overhead (%sys) is consistently higher than 60, or if the percentage of time waiting for physical I/O (%wio) is greater than 0, you may be paging excessively.

Excessive paging happens when you run out of memory, so that pages of executing processes must be paged out to disk memory. To correct the problem, you must free up some memory. Some ways to do this are:

- Reduce the size of the system buffer cache. System buffers occupy a large portion of the kernel. The number of buffers configured in the system is displayed when you boot the system, or can be viewed in the */usr/adm/messages* file. The **sar -b** report shows how the buffer cache is being used on your system; see “Tuning disk I/O parameters” (page 872) for more information about adjusting the size of the buffer cache.
- Reduce the number of STREAMS buffers by tuning the **NBLK<sub>n</sub>** parameters, especially if you are not using networking services or other applications that use STREAMS heavily. See “STREAMS data” (page 922) for more information.
- Look for other kernel tables that may be unnecessarily large. If the peak number of entries shown as being used on the **sar -v** report are consistently much less than table size, you can decrease table sizes and regain some memory. As a general rule, none of these tables should be tuned more than 15% higher than the highest level of usage. The entries in the process table (controlled by the **NPROC** tunable parameter) are fairly large, so reducing this tunable may help considerably. If you reduce the value of **NPROC**, you can also reduce the size of **NREGION** that controls the size of the region table to a value slightly more than  $3 \times \text{NPROC}$ .
- Run fewer graphical clients or choose versions of the clients that do not use the Motif libraries. See “Reducing memory dedicated to graphical clients” (page 880) for more information.

- Check that processes are not allocating unnecessarily large shared memory regions or leaving unused shared memory regions allocated when they exit abnormally. Use the `ipcs(ADM)` command to check the status of shared memory regions; use the `ipcrm(ADM)` command to removed an unused shared memory region from the system. You can use `crash(ADM)` to determine which processes are using shared memory.
- Add more memory to the configuration.
- Schedule large jobs to be run at off-hours or move users to another machine.
- Tuning the kernel to increase the value of `GPGSLO` and `GPGSHI` may help reduce the impact of excess paging if the other solutions cannot be implemented. `GPGSLO` is normally set to about 1/16 of pageable memory and `GPGSHI` is normally set to about 1/10 of pageable memory. Setting these tunables to higher values makes the paging daemon more active, which helps reduce the thrashing (excessive paging) that occurs when memory is extremely tight. See "Processes, memory management, and swapping" (page 915) for a description of these parameters.
- A few pseudo-drivers can be removed to save space, but because this is risky and the total savings are usually under 100 kilobytes, instructions have been omitted from this book.

## Kernel tables are too small

---

The kernel contains a number of tables that contain control information for executing processes. If these tables are too large, the kernel is unnecessarily large and you may have inadequate user space for executing processes. If these tables are too small, processes may fail or, in some cases, the system may panic.

When a table overflows, the kernel prints out a message such as the following:

```
CONFIG: timeout - Timeout table overflow (NCALL = number exceeded)
```

This identifies the table that has overflowed, the tunable parameter that controls the size of that table (`NCALL` in this case), and the current value of that tunable parameter.

The Table 2-2 summarizes the error messages that are produced when tables overflow, the tunable parameter to adjust to change the size of the table, and any tools that may be available to view usage statistics for the table.



**Table 2-2 Table overflow error messages**

| CONFIG: <i>text</i> ( <i>tunable</i> = <i>number</i> exceeded) |             |               |
|----------------------------------------------------------------|-------------|---------------|
| Text                                                           | Tunable     | Tools         |
| aio_breakup - AIO buffer table overflow                        | NAIOBUF     |               |
| aio_breakup - AIO request table overflow                       | NAIOREQ     |               |
| aio_memlock - AIO process table overflow                       | NAIOPROC    |               |
| aio_setlockauth - AIO lock table overflow                      | NAIOLOCKTBL |               |
| allocreg - Region table overflow                               | NREGION     |               |
| dk_name - Diskinfo table overflow                              | NDISK       |               |
| dosiread - DOS inode table overflow                            | DOSINODE    |               |
| event - Event channel full                                     | EVDEVSPERQ  |               |
| event - Event table full                                       | EVDEVS      |               |
| falloc - File table overflow                                   | NFILE       | sar -v        |
| hsiread - High Sierra inode table overflow                     | HSNINODE    |               |
| hsiread - High Sierra extent table overflow                    | HSNEXTENT   |               |
| id - Out of STREAMS                                            | NQUEUE      | crash strstat |
| ifreeget - Inode table overflow                                | NINODE      | sar -v        |
| MPX timeout - Timeout table overflow                           | NCALL       |               |
| newproc - Process table overflow                               | NPROC       | sar -v        |
| s5iread - <i>filesystem_type</i> inode table overflow          | NINODE      | sar -v        |
| sp: spclose - Cannot allocate STREAMS block                    | NBLK????    |               |
| string: Configuration buffer full                              | MAX_CFGSIZE |               |
| stropen1 - Out of streams                                      | NSTREAM     | crash strstat |
| stropen2 - Out of streams                                      | NSTREAM     | crash strstat |
| swapdel - Total swap area too small                            | MINASMEM    |               |
| timeout - Timeout table overflow                               | NCALL       |               |
| xsd_alloc - XENIX semaphore table overflow                     | XSEMMAX     |               |
| xtinit - Cannot allocate xt link buffers                       | NUMXT       |               |
| <i>kernel_function</i> - Out of clists                         | NCLIST      |               |

## Tuning disk I/O parameters

Many performance problems are caused by disk I/O bottlenecks. To correct these problems, you may need to rearrange the filesystems and swap areas on your disks, add more disks to your configuration, or map out bad areas of the disk so the system does not attempt to access them. See "Disk problems" (page 942) for information about analyzing and correcting disk I/O bottlenecks.

Disk performance may also be improved by tuning the parameters that control the system buffer cache. All disk I/O that uses the filesystem uses the buffer cache as an intermediate storage area. Most UNIX systems are I/O intensive, so assisting the I/O system with some tuning provides large

benefits for users, especially when running I/O intensive applications. The system buffer cache also occupies a large portion of kernel memory, so if you need to reduce the size of the kernel or are running applications that are not I/O intensive, you may improve performance by reducing the size of the buffer cache.

If you have sufficient memory, you may be able to improve I/O performance by modifying the following kernel parameters. If you are trying to reduce the size of the kernel, tuning the values of these parameters down may be a good starting place.

#### NBUF (and MAXBUF)

NBUF controls the amount of memory the system dedicates to the system buffer cache that caches disk read and write operations. Each caching buffer occupies 1K of RAM plus a bit of memory for the buffer header. Begin tuning by running `tunesh(ADM)`. `tunesh` tunes the system according to the amount of RAM as follows:

8 Mb or less            200 buffers

16 Mb or less          300 buffers

More than 16 Mb    NBUF = 0, so approximately 10% of RAM is reserved for the system buffer cache.

The number of buffers configured is displayed when you boot the system with a line similar to the following:

```
kernel: i/o bufs = xxxk.
```

This information is also recorded in the `/usr/adm/messages` file. After running the system with this value for a while, you can study the `sar -b` report to understand how the buffer cache is being used on your system. Usually you want the read cache hit rate ("`%rcache`" on the `sar -b` report) to be between 90% and 95%, and the write cache hit rate ("`%wcache`" on the `sar -b` report) to be between 65% and 85%. Theoretically, a cache hit rate of 100% is ideal, but in most cases sizing the buffer cache to achieve 100% hit rates uses so much memory that it causes performance problems in other areas. To determine the optimal buffer cache size for your system, adjust NBUF slightly and monitor the performance on your system. See "sar -b" (page 891) for more information.

Use the following guidelines when adjusting the size of the buffer cache:

- NBUF should usually have a value that is 10-25% of your system RAM. The tradeoff is that using more RAM for buffering leaves less RAM for user programs and may therefore increase paging.
- Adjust the size of the buffer cache in small increments, and monitor system performance carefully to see if additional adjustments are warranted.
- On most systems, the kernel should not exceed 16 Mbytes. Do not set NBUF so high that the kernel size exceeds this value unless you have a disk controller that supports DMA operations above 16 Mbytes. If such a controller is configured, the "%adapter fts = d" line will appear on the boot screen.

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NHBUF       | The NHBUF parameter controls the number of hash buckets used to hash the buffer cache. Hash buckets are arrays of buffers that are used to speed up searches in the buffer cache. Check the hardware screen that is displayed at boot time for the size of the buffer cache and set NHBUF to a power of 2 that is approximately 1/4 that size.                                                                                                            |
| S5CACHEENTS | The S5CACHEENTS parameter controls the size of the <code>namei</code> cache that is used to speed the time searching through all directories in a path name when a file is first accessed. This caching speeds up the time required to locate a file. A good starting point for S5CACHEENTS is 3*NINODE. If many users are accessing the same set of files, the <code>namei</code> cache can be smaller than if the users are accessing different files.  |
| S5HASHQS    | S5HASHQS controls the size of the hash table that is used with the <code>namei</code> cache. This hash table works much like the buffer cache hash table discussed above. S5HASHQS should be set to a prime number that is roughly 1/4 the value of S5CACHENTS.                                                                                                                                                                                           |
| S5OFBIAS    | S5OFBIAS controls the length of time the system spends searching the <code>namei</code> cache for a free entry before it takes an entry that is in use (thus deleting the entry from the cache). The larger the value, the longer the search, but the more likely that an unused entry will be found rather than displacing a current entry. Typically, S5OFBIAS is set to 1/10 the value of S5CACHEENTS, and it must not be any larger than S5CACHEENTS. |

For information about interpreting the `sar(ADM)` reports that report information to use when tuning the I/O subsystem, see “`sar -b`” (page 891), “`sar -n`” (page 895), and “`sar -d`” (page 893).

## Optimizing serial I/O

This section covers some configuration changes that you can make to optimize serial throughput using the SCO serial driver on standard serial ports and non-intelligent cards. In particular, by changing your configuration, you can use the 16550 FIFO support in the SCO serial driver to obtain sustained data transfer rates greater than 9600 bps.

To support higher transfer rates, you should do the following.

**NOTE** The optimum configuration varies from system to system; you might need to experiment to determine the configuration that works best for your system.

1. Increase the value of the `TTHOG` kernel parameter.

`TTHOG` is the maximum size of the raw queue of the `tty` driver. Increasing the value of this parameter allows more unprocessed characters to be retained in the `tty` buffer, which may prevent input characters from being lost if the system is extremely busy.

The default value of `TTHOG` (256) is too small for high speed throughput. We recommend that you start by increasing `TTHOG` to 2048. Some applications might need a higher value, such as 4096.

**NOTE** If you increase the value of `TTHOG`, you must also increase the value of the `NCLIST` kernel parameter (see step 2).

2. Increase the value of the `NCLIST` kernel parameter.

Each serial port consumes 64-byte cblocks until it has used the number of bytes specified by `TTHOG`. If you increase the value of `TTHOG` to 2048, each serial port can use up to 32 cblocks. Therefore, you should increase the number of cblocks (by increasing the `NCLIST` kernel parameter) by  $32 \times (\text{number of serial ports})$ . If you increase `TTHOG` to 4096, you should increase `NCLIST` to  $64 \times (\text{number of serial ports})$ .

3. Change the default values in the `sio_fifoct1` kernel array.

For serial ports using the 16550AFN UART chip, you can change the receive interrupt trigger level from 14 to 8 by changing the default values in the `sio_fifoct1` array. This causes the 16550 to interrupt sooner, so that a longer interrupt latency can be tolerated without character loss.

Possible values for the interrupt trigger level are 14, 8, 4, and 1. A value of 8 is a good starting point; however, some applications might require a value of 4 or even 1. Lower interrupt trigger levels are appropriate when the kernel has a high maximum “interrupt latency”. (The interrupt latency is the amount of time it takes the kernel to begin servicing the interrupt.)

### *Increasing the value of TTHOG and NCLIST*

---

To increase the value of TTHOG and NCLIST:

1. Log in as *root* and bring the system down to system maintenance (single-user) mode. See the section “Entering system maintenance mode directly” in the chapter on starting and stopping the system in *Administering Operating System Services*.

2. Enter the following commands:

```
cd /etc/conf/cf.d
./configure
```

Δ **sysadmsh** users select: System ⇌ Configure ⇌ Kernel ⇌ Parameters

3. Select “2. Character Buffers” from the menu.
4. Press (Return) until you see the TTHOG prompt.
5. At the TTHOG prompt, enter the new value for TTHOG (either 2048 or 4096, depending on your application).
6. Press (Return) until you return to the **configure** menu.
7. Select “2. Character Buffers”.
8. Press (Return) until you see the NCLIST prompt.
9. At the NCLIST prompt, enter the new value for NCLIST.

Use the following formulas to calculate the new value for NCLIST:

**TTHOG = 2048**    new NCLIST value = old NCLIST value + 32\*(number of serial ports)

**TTHOG = 4096**    new NCLIST value = old NCLIST value + 64\*(number of serial ports)

10. Press (Return) until you return to the **configure** menu.
11. Enter “q” to quit **configure**.
12. Enter “y” to update the system configuration files.

The TTHOG and NCLIST kernel parameters are now updated. For these changes to take effect, relink the kernel; see “Relinking the kernel” (page 879).

## Changing the `sio_fifoctl` array

The `sio_fifoctl` array is initialized in the `/etc/conf/pack.d/sio/space.c` file. You can edit `space.c` with the `vi(C)` editor.

To change the `sio_fifoctl` array in the `space.c` file:

1. Determine the elements of the `sio_fifoctl` array.

The elements of the `sio_fifoctl` array correspond to the minor numbers of the non-modem control ports and the minor number minus 128 of modem control ports:

### non-modem control port

To determine the minor number for a non-modem control port, display a long listing of that port in the `/dev` directory. The minor number is the integer immediately before the date. Use this number for the `sio_fifoctl` array index.

For example, to determine the minor number for the `tty1a` control port, enter:

```
l /dev/tty1a
```

You see:

```
crw-rw-rw- 1 bin bin 5, 0 Jan 22 1992 /dev/tty1a
```

In this case, the minor number is 0; so you must modify the first array element (`sio_fifoctl[0]`).

### modem control port

To determine the minor number for a modem control port, display a long listing of that port in the `/dev` directory and subtract 128 from the minor number (the integer immediately before the date). Use this number for the `sio_fifoctl` array index.

For example, to determine the minor number for the `tty2A` modem control port, enter:

```
l /dev/tty2A
```

You see:

```
crw-rw-rw- 1 bin bin 5,136 Jan 22 1992 /dev/tty2A
```

In this case, the minor number is 136; 136 minus 128 is 8, so you must modify the 9th array element (`sio_fifoctl[8]`).

**NOTE** The minor number for `tty2A` might be different on your system.

2. Enter the following commands:

```
cd /etc/conf/pack.d/sio
cp space.c space.c.old
```

3. Edit *space.c* to make the desired changes to the **sio\_fifoc1** array:<sup>1</sup>

```
146 /*
147 * 16550 FIFO control by minor dev: the high two bits are the RX fifo
148 * trigger level, and the low four bits are the size-1 to use for the
149 * TX fifo. Statically initialize the array here, so that buffering
150 * of individual channels can be disabled, e.g., for serial mouse.
151 *
152 * NB: The format of this array must not be changed, since it is edited
153 * by the 'mkdev mouse' script, which relies on having one entry per line.
154 */
155 unsigned char sio_fifoc1[CHANNELS] = {
156 FCR_Rx14 | 15, /* <- This is the 1st array element, sio_fifoc1[0] */
157 FCR_Rx14 | 15, /* <- This is the 2nd array element, sio_fifoc1[1] */
158 FCR_Rx14 | 15, /* <- This is the 3rd array element, sio_fifoc1[2] */
159 FCR_Rx14 | 15,
160 FCR_Rx14 | 15,
161 FCR_Rx14 | 15,
162 FCR_Rx14 | 15,
163 FCR_Rx14 | 15,
164 FCR_Rx14 | 15, /* <- This is the 9th array element, sio_fifoc1[8] */
165 FCR_Rx14 | 15,
166 FCR_Rx14 | 15, /* FCR_Rx1, FCR_Rx4, FCR_Rx8, and FCR_Rx14 are */
167 FCR_Rx14 | 15, /* defined in <sys/siohw.h>. They set the interrupt */
168 FCR_Rx14 | 15, /* trigger level to 1, 4, 8, and 14 respectively. */
169 FCR_Rx14 | 15, /* */
170 FCR_Rx14 | 15, /* 14 is the default trigger level */
171 FCR_Rx14 | 15, /* */
172 FCR_Rx14 | 15, /* The *15* means to use a size of 14 for the TX */
173 FCR_Rx14 | 15, /* fifo. */
174 FCR_Rx14 | 15,
175 FCR_Rx14 | 15,
176 FCR_Rx14 | 15,
177 FCR_Rx14 | 15,
178 FCR_Rx14 | 15,
179 FCR_Rx14 | 15, /* <- This is the 24th array element, sio_fifoc1[23] */
180 };
```

---

1. The comments to the left of the "FCR\_Rx14 | 15" lines do not appear in the *space.c* file; you do not need to add them.

4. Modify the appropriate line in the *space.c* file. For example:

**sio\_fifoct1[0]** (the usual entry for tty1a/A)

- To change the interrupt trigger level to 8, change line 156 to:

```
FCR_Rx8 | 15,
```

- To change the interrupt trigger level to 1, change line 156 to:

```
FCR_Rx1 | 15,
```

**sio\_fifoct1[8]** (the entry for tty2a/A, as calculated in step 1)

- To change the interrupt trigger level to 8, change line 164 to:

```
FCR_Rx8 | 15,
```

- To change the interrupt trigger level to 1, change line 164 to:

```
FCR_Rx1 | 15,
```

The changes made to *space.c* do not take effect until you relink the kernel; see “Relinking the kernel” (this page).

**NOTE** If you are using a port as a mouse port, that element of the **sio\_fifoct1** array is zeroed. For example, if you are using tty1a as a mouse port, line 156 in your *space.c* file looks like this:

```
0,
```

## Relinking the kernel

To relink the kernel:

1. Log in as *root* and bring the system down to system maintenance (single-user) mode. See the section “Entering system maintenance mode directly” in the chapter on starting and stopping the system in *Administering Operating System Services*.
2. Enter the following commands:
 

```
cd /etc/conf/cf.d
./link_unix
```
3. At the prompt to specify that the new kernel boot by default, enter “y”.
4. At the prompt to rebuild the kernel environment, enter “y”.
5. Enter:

```
init 6
```

This reboots the system with the new kernel.



## *Reducing memory dedicated to graphical clients*

---

If you are short of memory, you can reduce the amount of user memory required by simplifying the Desktop environment. Many of the graphical clients supplied with SCO Open Desktop/SCO Open Server are very large, especially those that use the Motif libraries, so running a smaller subset of these applications can significantly reduce your memory requirements. Users with tight memory configurations can consider changes such as the following:

- Do not use the session manager, **scoession**, to start up your clients. Instead, create a **.startxrc** file in your home directory and specify the clients to run in that file. Also reconfigure your root menu so it does not run **scoession** when you log out.

**scopald** must be the first client started in the **.startxrc** file if you want color support; it should not be started in the background. **mwm**, the Motif Window Manager, also usually runs in the foreground; other clients will probably be started in the background. See the *Graphical Environment Administrator's Guide* for more information about **.startxrc** files.

- Disable **scologin** by logging in as root at the console login prompt and issuing the **/etc/scologin disable** command. This saves about 3 processes and the associated memory. With **scologin** disabled, you will need to login at the regular UNIX system prompt and type **startx** to begin the graphical clients. See **scologin(X)** for more information.
- Disable the Desktop client, **xdt3** if you do not need to click and drag icons for tasks. You can configure the **mwm** root menu to list clients that you want to start often, or you can start clients with the **xterm** or **scoterm** command line. See the *Graphical Environment Administrator's Guide* for more information about adding clients to the root menu.
- Use **xterm** rather than **scoterm** for the terminal emulator. **scoterm** includes some functionality that is not available in **xterm**, but it is also Motif-based and consequently consumes about twice as much memory as **xterm**.
- Access **mail** from the command line rather than using **scomail**.
- Use **vi(C)** to edit files and disable **scoedit**, which is a Motif application.
- If you do not access the online documentation often, shutdown **scohelp** when you are done reading it. Manual pages can be accessed with the **xman(X)** or **man(C)** command, although you will not have the hyperlinks to use to jump between pages. Initializing **scohelp** takes some time because several large files must be loaded, so it may be worth the memory to leave **scohelp** running when you expect to access online help.
- The **xclock** client is fairly small and does not have a significant impact on system performance.

## Chapter 3

# Tools reference

---

The operating system includes tools that measure performance, reconfigure the system for better performance, and analyze other system problems that arise. The system administrator can use these tools to locate problem areas. The tools described in this section are:

|                  |                                                                                                                                                                                                                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>sar</b>       | samples the state of the system and provides reports on various system-wide activities.                                                                                                                                                                                                                                         |
| <b>swap</b>      | with the <b>-l</b> option, reports on the amount of available swap space. Use the <b>-a</b> option to configure additional swap devices.                                                                                                                                                                                        |
| <b>crash</b>     | displays various kernel structures. The discussion here concentrates on using <b>crash</b> to view how STREAMS resources are being used, to see the current value of many tunable parameters, and to study the state of the system at the time of a system PANIC. <b>crash</b> has many other uses that are not discussed here. |
| <b>ps</b>        | reports on processes currently executing.                                                                                                                                                                                                                                                                                       |
| <b>timex</b>     | reports on system resource usage during the execution of a command or program.                                                                                                                                                                                                                                                  |
| <b>configure</b> | a menu-driven program used to modify the value of tunable parameters that are included in the Operating System Services component of this product.                                                                                                                                                                              |
| <b>fsck</b>      | checks the integrity of a filesystem and attempts to fix any corruption it finds.                                                                                                                                                                                                                                               |
| <b>id tune</b>   | used to modify some tunable parameters that cannot be modified with <b>configure</b> .                                                                                                                                                                                                                                          |
| <b>integrity</b> | checks for missing system files or incorrect permissions on system files and directories.                                                                                                                                                                                                                                       |

**tunesh**      an automatic configuration tool that adjusts kernel parameters based on information you supply about your configuration

## *configure(ADM) — reallocate kernel resources*

---

The **configure** utility is a menu-driven program that presents each tunable kernel parameter and prompts for modification. After modifying kernel parameters, you must relink the kernel by invoking **link\_unix(ADM)**, reboot, and test the new kernel. To change any kernel parameter, do the following:

1. Reboot and enter single-user (maintenance) mode.
2. Enter the following commands to start the **configure** program:

```
cd /etc/conf/cf.d
./configure
```

Δ **sysadmsh** users select: System ⇄ Configure ⇄ Kernel ⇄ Parameters

3. The **configure** menu is displayed. The parameters are grouped by category. The meanings of the parameters are discussed in Tunable system parameter descriptions (page 907).

Choose a category by entering the number preceding it. The resources in that category are displayed, one by one, each with its current value. Enter a new value for the resource, or to retain the current value, simply press (Return). After all the resources in the category are displayed, **configure** returns to the category menu prompt. Return to the Main Menu to choose another category or exit **configure** by entering "q".

4. After you finish changing parameters, you must link them into a new kernel. Enter the following command:

```
./link_unix
```

Δ **sysadmsh** users select: System ⇄ Configure ⇄ Kernel ⇄ Rebuild

This assembles each of the kernel modules into a new kernel, which must now be installed. Linking can take a few minutes.

5. Boot the new kernel with the following command:

```
/etc/shutdown
```

Δ **sysadmsh** users select: System ⇄ Terminate

A "Boot:" prompt appears. When you press (Return) to reboot the system, the new kernel is loaded and run.

If problems exist with the new kernel, reboot */unix.old* by entering "unix.old" at the "Boot:" prompt.

## Using the configure command line

---

**configure** also has a command-line interface suitable for use by application developers. For instance, a database developer who finds that 70 files rather than 50 files need to be locked simultaneously may provide a shell script to perform the reconfiguration. To find the current value of any configurable resource using the command-line interface, enter:

```
./configure -y RESOURCE
```

Here *RESOURCE* is the name of the tunable parameter (in uppercase). To change the value of any resource from the command line, enter:

```
./configure RESOURCE=value
```

## Overriding configure limit warnings

---

The **configure** utility notifies you if you exceed certain limits on values for kernel parameters. This prevents you from accidentally exceeding a value, which could severely degrade performance in some cases. The **configure** menu program verifies that you do mean to override the limits. You can also use the **-o** option as described on the **configure(ADM)** manual page to override a configuration limit. The override option only works if you are specifying a parameter on the command line; you cannot use it with the menu.

## crash(ADM) — view memory

---

The **crash(ADM)** utility allows you to analyze the core image of the operating system. It is frequently used in postmortem analysis of a system PANIC, but can also be run on an active system. **crash** is not primarily an administrative tool; it is most useful for debugging the kernel after installing drivers and other kernel-level code. Nonetheless, **crash** provides some information that can help you monitor the performance of your system and troubleshoot system problems that arise. This documentation provides information about **crash** reports related to these topics; see the **crash(ADM)** manual page for full information about using **crash**.

## Initializing crash

---

You must be logged in as *root* to run **crash**. The full command to begin the **crash** session is:

```
/etc/crash [-d dumpfile -n namelist] [-w outputfile]
```

The dumpfile is */dev/mem* for an active system (the default), or the name of the file that contains the memory dump you saved when rebooting the system after a system PANIC. *namelist* is the name of the bootable kernel image, usually */unix* unless you booted the system under a different name.

When **crash** is initialized, you see the following display:

```
dumpfile= /dev/mem, namelist= /unix, outfile= stdout
>
```

The ">" is the **crash** prompt; enter the **crash** function you want followed by any options you need and (Return). Note the following:

- The **-w** option redirects the output of the command to the file specified. For example, **-w var\_stats** writes the output of **crash var** to the *var\_stats* file. You can use **-w** on the **crash** command line to redirect all **crash** output to a file, or you can use **-w** on the function line to redirect the output of that particular function to a file.
- Any **crash** function can be piped to any shell function. For instance, the command **var ! cat >> file** takes the output of the **crash var** function and appends it to the end of *file*; **var ! lp** redirects the output of this same command to the line printer. Note that the exclamation point (!) serves as the first pipe symbol from within **crash**; use " | " thereafter.
- Use the **q** function to end the crash session.

The **crash stat** command is a good starting place when analyzing a postmortem dump. **stat** lists offsystem statistics (such as the system name and time of PANIC) that you can use to verify that you are looking at the right dump. The **crash panic** command will usually contain the last PANIC message that was displayed on the console.

### ***crash strstat (statistics about STREAMS buffers)***

Use **crash strstat** to view current usage statistics on the STREAMS buffers for your system. This information can be used to select correct values for the **NBLK<sub>n</sub>** tunable parameters and other tunable parameters discussed in "STREAMS data" (page 922). The following example shows a sample session using the **crash strstat** utility. The listings are associated with the following parameters:

```
streams NSTREAM
queues NQUEUE
data block size n NBLKn
```

The listing "message blocks" is a composite number based on the total block allocation; it does not correspond to a kernel parameter. The **NBLK<sub>n</sub>** data blocks are also known as STREAMS buffers.

**Sample crash strstat output**

```
/etc/crash
dumpfile = /dev/mem, namelist = /unix, outfile = stdout
> strstat
> ITEM CONFIG ALLOC FREE TOTAL MAX FAIL
streams 168 84 84 1190 86 0
queues 624 380 244 2621 388 0
message blocks 1455 155 1300 1332119 323 0
data block totals 1164 155 1009 1079923 322 198
data block size 4 256 0 256 28571 19 0
data block size 16 168 2 166 10540 52 0
data block size 64 296 29 267 831712 90 0
data block size 128 168 87 81 152605 139 185
data block size 256 104 29 75 18593 83 13
data block size 512 72 8 64 31949 10 0
data block size 1024 48 0 48 2286 8 0
data block size 2048 40 0 40 3601 6 0
data block size 4096 12 0 12 66 1 0

Count of scheduled queues: 0
> quit
```

Table 3-1 defines the column headings used by the `strstat` display.

**Table 3-1 STREAMS display headings**

| Heading | Description                                                                                       |
|---------|---------------------------------------------------------------------------------------------------|
| CONFIG  | number of items currently configured                                                              |
| ALLOC   | number of items currently allocated                                                               |
| FREE    | number of items available for allocation. This number is the difference between CONFIG and ALLOC. |
| TOTAL   | number of attempted allocations since system startup                                              |
| MAX     | highest number of items allocated at one time since system startup.                               |
| FAIL    | number of failures due to insufficient free items since system startup                            |

A non-zero value in the “FAIL” column indicates that a request was made for a buffer and that no buffers of that size were available. STREAMS failures cause problems with TCP/IP, degrade system performance and, in extreme cases, can cause your system to lock up. In this situation, use the `configure(ADM)` command to increase the number of buffers.

## *var (values of tunable parameters)*

---

The **crash var** command lists the values of the tunable parameters that are listed in the **var** structure. These same values are listed when you run **configure(ADM)** or **sysdef(ADM)**, but the **crash** listing is useful when you want to check a value when you are running **crash**.

## *fsck(ADM) — check and repair filesystems*

---

The **fsck(ADM)** program should be run on all filesystems after any system crash or abnormal system termination. **fsck** checks filesystem integrity. Because filesystems are not unmounted gracefully when the system crashes, the structure of the filesystem itself as well as user, system, and application files that were in the process of being updated can be damaged. In most cases, **fsck** fixes any filesystem problems; in some cases, **fsck** may be unable to fix a file, but can recover some portion of the file in the *lost+found* directory of the filesystem. The filesystem being checked must be unmounted when you run **fsck**; the system must be in single-user mode when running **fsck** on the *root* filesystem. If the filesystem is corrupted to a point where **fsck** is unable to recover it, you may be able to fix the filesystem manually with **fsdb(ADM)**. See “Repairing a filesystem when **fsck** stops at size check” (page 951) for information. Otherwise, you may need to recreate the filesystem and restore the data from a backup.

Filesystems should always be checked after a system crash before they are mounted. Minor filesystem corruption can become widespread quickly if you continue to read and write a corrupted filesystem. **fsck** can be run with the **-y** option so that it will automatically attempt to fix any filesystem corruption it finds. **fsck -y** runs when rebooting the system if the appropriate string is in the files that control the booting process are set up appropriately:

| Filesystem | String        | File                 |
|------------|---------------|----------------------|
| root       | FSCKFIX = YES | /etc/default/boot    |
| all others | fsckflags = y | /etc/default/filesys |

Without these lines, **fsck** reports each error it finds and waits for you to respond before it attempts to fix it during reboot. If these lines are not present and you have **AUTOBOOT=YES** in */etc/default/boot* (default), the system will not automatically reboot if any filesystems suffered any damage at all from the crash.

## *id tune(ADM) — reallocate kernel resources*

---

The **id tune(ADM)** command modifies parameters that are stored in the */etc/conf/cf.d/mtune* file, much as the **configure(ADM)** command is used to modify standard kernel parameters. You should never edit the *mtune* file directly.

To change the value of a resource, enter the following commands:

```
cd /etc/conf/cf.d
../bin/idtune RESOURCE=value
```

Here **RESOURCE** is the name of the tunable parameter (in uppercase) and **value** is the new value for that parameter. After using the **id tune** command, you should relink the kernel and shut down the system as with the **configure** command.

## *integrity(ADM) — check system file permissions*

---

The **integrity(ADM)** program compares the entries of the File Control database against the actual file permissions on the system. If your system is configured with the High security defaults and **integrity** finds inconsistencies, use the **fixmog(ADM)** command to fix the file permissions.

Run **integrity** as follows:

```
/tcbin/integrity -m -e > int.report
```

Print the file *int.report* and examine it. **integrity** reports files and directories that are missing or have incorrect permissions or ownership. Here are sample messages generated by **integrity**:

```
/etc/utmp (entry 83) is wrong.
 Owner is root, should be bin.
 Group is root, should be bin.
 Mode is 0644, should be 0664.
/usr/spool/lp (entry 233) is wrong.
 Group is bin, should be lp.
 Mode is 0755, should be 0070.
/etc/inittab (entry 71) is wrong.
 Type is d. should be r.
/usr/lib/mkuser/csh (wildcard entry 216) is wrong.
 Owner is bin, should be root.
 Mode is 0700, should be 0750.
```

The owner, group, and mode refer to the file permissions. The file types “d” and “r” refer to directory, and regular file, respectively. Missing files should be replaced by restoring them from backups. Permission and some “type” problems can be fixed with the **fixmog** utility. All errors found during the



integrity check are packaged as audit records that show the audit event as a Database Event in the audit trail.

**NOTE** Some files may be listed as missing in a correctly configured system, such as one of the pair `/usr/lib/cron/at.allow` and `/usr/lib/cron/at.deny`.

## *ps(C) — check process activity*

---

The `ps(C)` command obtains information about active processes. This command gives a “snapshot” picture of what processes are executing, which is useful when you are trying to identify what processes are loading the system. Things will probably change by the time the output appears; however, the entries that you should be interested in are **TIME** (minutes and seconds of CPU time used by processes) and **STIME** (time when the process first started). The following example contains sample output from the `ps -af` command.

### Sample output from `ps -af`

|  | UID     | PID   | PPID  | C  | STIME    | TTY | TIME | COMMAND              |
|--|---------|-------|-------|----|----------|-----|------|----------------------|
|  | root    | 22247 | 21299 | 0  | 16:54:41 | T1  | 0:01 | mscreen -n 2         |
|  | johnson | 22246 | 22079 | 0  | 16:52:53 | T0  | 0:04 | vi file2             |
|  | root    | 22285 | 22247 | 0  | 16:56:04 | T1  | 0:06 | mscreen -n 2         |
|  | root    | 22284 | 22247 | 0  | 16:56:04 | T1  | 0:00 | mscreen -n 2         |
|  | markham | 22274 | 22271 | 0  | 16:55:09 | p1  | 1:05 | rlogin colossus      |
|  | markham | 22271 | 22243 | 0  | 16:55:07 | p1  | 1:05 | rlogin colossus      |
|  | forbin  | 22304 | 21003 | 0  | 17:02:51 | 003 | 0:00 | /usr/bin/mail kuprin |
|  | fisher  | 22298 | 18505 | 2  | 17:02:29 | 011 | 3:28 | nethack              |
|  | root    | 22305 | 1327  | 10 | 17:02:57 | T2  | 0:00 | ps -afe              |

**NOTE** If you booted your system from a file other than `/unix` (such as `/unix.old`), you must specify the name of that file with the `-n` option to `ps`. For example, `ps -ef -n unix.old`.

See “Runaway processes, stopping” (page 974) for information about using the `ps` command to perform such tasks as locating “runaway” processes (one that uses progressively more system resources over a period of time while you are monitoring it). You can also use `ps` to find processes that take a very long time to execute; you can consider using `cron(C)` to execute such jobs during an idle time of the day.

## ***sar(ADM) — system activity reporter***

---

**sar(ADM)** provides information that can help you understand how system resources are being used on your system. This information can help you solve and avoid serious performance problems on your system.

Table 3-2 summarizes the **sar** options:

**Table 3-2 sar options**

| Option | Reports on:                                 |
|--------|---------------------------------------------|
| -a     | file access system routines                 |
| -b     | buffer cache activity                       |
| -c     | system calls                                |
| -d     | disk activity                               |
| -m     | IPC message and semaphore activity          |
| -n     | name cache                                  |
| -p     | paging activity                             |
| -q     | average length of run and swap queues       |
| -r     | unused memory pages and disk blocks         |
| -u     | CPU utilization                             |
| -v     | process, inode, file, and lock table status |
| -w     | system swapping and switching activity      |
| -y     | TTY device activity                         |
| -A     | summary of all reports                      |

The individual **sar** outputs are discussed in alphabetical order later in this chapter.

### ***How sar works***

---

Internal activity is measured by a number of counters contained in the kernel. Each time an operation is performed, an associated counter is incremented. The **sar(ADM)** utility generates reports based on the raw data gathered from these counters. **sar** reports can be used to diagnose system problems. The two most critical areas to monitor are memory and CPU usage. The functions monitored by **sar** are discussed in the subsections that follow, including analysis of sample **sar** output. **sar** can either gather system activity data “live” or extract information collected in data files created by **sadc** (System Activity Data Collector).

By default, the following **crontab** entry is installed in **sys** file in the **/usr/spool/cron/crontabs** directory:

```
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

The **sa1** entries produce records every 20 minutes during working hours and hourly otherwise. The **sa2** entry writes a daily average report into the same file every hour during work hours. You can modify these entries as desired. The output files are in binary format (for compactness) and are stored in **/usr/adm/sa**. The filenames have the format **sadd**, where **dd** is the day of the month. (See the **crontab(C)** and **sar(ADM)** manual pages for more information on modifying the default **crontab** file.)

## Running sar

---

The basic syntax of the **sar(ADM)** command is as follows:

```
sar [-option] [file]
```

where:

**option** one of the options described in the following sections

**file** name of the file for the day whose statistics you want to view. For example, to view the **sar -v** report for the tenth day of the most recent month, the command is:

```
sar -v -f /usr/adm/sa/sa10
```

You can also run **sar** in “real time”. To do this, specify the sampling interval in seconds. For example, to take a sample every 15 seconds, the command is:

```
sar -v 15
```

As shipped, the system allows any user to run **sar** in real time. However, the files in the **/usr/adm/sa** directory are readable only by **root**. Change the permissions on the files in that directory to allow other users to access the **sar** data.

The examples in this section are not intended to represent benchmarks; they serve to illustrate how the output can be used. When tuning your system, it is recommended that you use a “benchmark” (a program used to evaluate the performance of a system) and have the system under normal load for your application.

## **sar -a — file access operations**

---

The **sar -a** option reports the use of file access operations. The operating system routines reported are as follows:

**iget/s**      number of files located by i-node entry per second

**namei/s**    number of filesystem path searches per second. "namei" calls "iget", so "iget/s" is always larger than "namei/s".

**dirbk/s**    number of directory block reads issued per second

An example of **sar -a** output, with a 30-second sampling interval, follows:

```

unix unix 3.2 2 i386 07/18/92
12:41:40 iget/s namei/s dirbk/s
12:42:10 4 1 3
12:42:40 2 1 1
12:43:10 5 2 3
Average 4 1 3

```

The larger the values reported, the more time the kernel spends accessing user files. This indicates how heavily programs and applications are using the filesystem(s). In general, if the ratio of "iget/s" to "namei/s" is greater than 5 and "namei/s" is greater than 30, it may indicate that your filesystem organization is inefficient. Refer to "Checking free space on filesystems" (page 947) and "Reducing disk fragmentation" (page 947) for ways to solve the problem.

## **sar -b — buffer activity**

---

The **-b** option reports the following buffer activity.

**bread/s**    average number of physical blocks read into the system buffers from the disk (or other block devices) per second

**lread/s**    average number of logical blocks read from system buffers per second

**%rcache**    fraction of logical reads found in buffer cache (100% minus the ratio of breads to lreads)

**bwrit/s**    average number of physical blocks written from the system buffers to disk (or other block devices) per second

**lwrit/s**    average number of logical blocks written to system buffers per second

**%wcache**    fraction of logical writes found in buffer cache (100% minus the ratio of bwrit/s to lwrit/s)

**pread/s**    average number of physical read requests per second

**pwrit/s**    average number of physical write requests per second

The entries that you should be most interested in are the cache hit ratios %rcache and %wcache, which measure the effectiveness of system buffering. If %rcache falls below 90, or %wcache falls below 65, it may be possible to improve performance by increasing the number of buffers. If your application is I/O-intensive and you have a large memory configuration, you may want to tune the buffer cache so that %rcache is around 95 and %wcache is around 80. The number of buffers in the system buffer cache are controlled by the NBUF tunable parameter. See "Tuning disk I/O parameters" (page 872) for more information.

Technically, 100% hit ratio for the read cache should mean better performance than the 90% recommended here, but getting that last 10% of hits often requires so much memory that performance suffers because of the lack of user memory for executing processes.

An example of `sar -b` output follows:

```
unix unix 3.2 2 i386 07/18/92
16:32:57 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
16:33:07 3 39 93 1 16 91 0 0
16:33:17 4 40 90 2 16 87 0 0
16:33:27 4 41 90 3 7 64 0 0
Average 4 40 91 2 13 84 0 0
```

This example shows that the buffers are not causing any bottlenecks, because all data is within acceptable limits.

## ***sar -c — system call activity***

---

The `-c` option reports system calls in the following categories:

|         |                                                                                                                                 |
|---------|---------------------------------------------------------------------------------------------------------------------------------|
| scall/s | all types of system calls per second, generally about 30 per second on a busy 4 to 6 user system                                |
| sread/s | read system calls per second                                                                                                    |
| swrit/s | write system calls per second                                                                                                   |
| fork/s  | fork system calls per second, about 0.5 per second on a 4 to 6 user system. This number increases if shell scripts are running. |
| exec/s  | exec system calls per second. If "exec/s" / "fork/s" is excessively large, look for inefficient \$PATH's.                       |
| rchar/s | characters (bytes) transferred by read system calls per second                                                                  |
| wchar/s | characters (bytes) transferred by write system calls per second                                                                 |

Typically, reads plus writes account for about half of the total system calls, although this varies greatly with the activities that are being performed by the system.

An example of **sar -c** output follows:

```

unix unix 3.2 2 i386 07/18/92
18:33:04 scall/s sread/s swrit/s fork/s exec/s rchar/s wchar/s
18:33:35 38 16 6 0.03 0.03 6089 1638
18:34:05 39 16 4 0.07 0.07 6123 1602
18:34:35 38 17 5 0.17 0.17 6042 1704
Average 38 16 5 0.09 0.09 6085 1648

```

This report is of interest mostly for programmers who are testing application programs. If **scall/s** is greater than 300 over an extended period of time, it may indicate inefficient code for applications that are running.

Note that the XENIX system **vmstat(C)** command that is also supported on SCO Open Desktop/SCO Open Server provides cumulative statistics about system calls executed (**vmstat -s**) and number of forks (**vmstat -f**).

## **sar -d** — block device activity

The **sar -d** option reports the activity of block devices.

|               |                                                                                                                                                               |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>device</b> | name of the block device(s) that <b>sar</b> is monitoring                                                                                                     |
| <b>%busy</b>  | percent of time the device was servicing a transfer request                                                                                                   |
| <b>avque</b>  | average number of requests outstanding during the period of time (measured only when the queue is occupied)                                                   |
| <b>r+w/s</b>  | number of read and write transfers to the device per second                                                                                                   |
| <b>blks/s</b> | number of 512 byte blocks transferred to the device per second                                                                                                |
| <b>await</b>  | average time in milliseconds that transfer requests wait idle in the queue (measured only when the queue is occupied)                                         |
| <b>avserv</b> | average time in milliseconds for a transfer request to be completed by the device (for disks this includes seek, rotational latency, and data transfer times) |

An example of **sar -d** is as follows:

```

unix unix 3.2 2 i386 02/18/92
13:46:28 device %busy avque r+w/s blks/s await avserv
13:46:58 wd-0 6 1.6 3 5 13.8 23.7
 f1-0 93 2.1 2 4 467.8 444.0
13:47:28 wd-0 13 1.3 4 8 10.8 32.3
 f1-0 100 3.1 2 5 857.4 404.1
13:47:58 wd-0 17 .7 2 41 .6 48.1
 f1-0 100 4.4 2 6 1451.9 406.5
Average wd-0 12 1.2 3 18 8.4 34.7
 f1-0 98 3.2 2 5 925.7 418.2

```

Using this information, you can see how often each device is in use, how many requests (on average) are waiting for the disk, and how quickly the disk can find and transfer the data. The above example was taken while transferring data from hard disk (wd-0) to floppy disk (f1-0). The large difference between the numbers for the two devices illustrates the performance advantages of hard disks.

Note that queue lengths and wait times are measured while the queue had something on it. If %busy is small, large queues and service times probably represent the periodic sync efforts by the system to ensure that altered blocks are written to the disk in a timely fashion.

The optimum setup is to keep the %busy number high and the avque number low. To do this, balance filesystems and swap areas across all disks and controllers or host adapters. See "Disk load balancing" (page 943) for more information.

### ***sar -m — interprocess communication***

---

The **sar -m** option reports on interprocess communication activities. Message and semaphore calls are reported as follows:

msg/s      number of message operations (sends and receives) per second  
sema/s      number of semaphore operations per second

An example of **sar -m** output follows:

```
unix unix 3.2 2 i386 02/18/92
15:16:58 msg/s sema/s
15:17:32 12.3 29.2
15:18:02 11.8 26.8
15:18:32 18.4 23.1
Average 14.2 26.4
```

If you are not running application programs that use messages and System V semaphores, these figures will all be equal to 0.00. If you are using these interprocess communications facilities and either msg/s or sem/s is greater than 100, the application is not using the system efficiently. See "Message queues" (page 919) and "Semaphores" (page 920) for information about tuning these parameters.

## **sar -n — name cache activity**

---

The **sar -n** option reports the name cache statistics.

|                |                                   |
|----------------|-----------------------------------|
| <b>c_hits</b>  | number of cache hits              |
| <b>cmisses</b> | number of cache misses            |
| <b>hit%</b>    | hit to miss ratio as a percentage |

An example of **sar -n** follows:

```

unix unix 3.2 2 i386 02/18/92

11:26:05 c_hits cmisses (hit %)
11:26:10 9 0 (100%)

```

The **namei** cache improves the time required to search a full pathname when first accessing a file. Generally, the higher the “hit %” field the better. If the “hit %” consistently falls below 90%, increase the value of **S5CACHEENTS** and associated parameters. See “Tuning disk I/O parameters” (page 872) and **S5CACHEENTS** (page 915) for more information.

## **sar -p — paging activity**

---

The **-p** option reports paging activity. The following page rates are recorded.

|                |                                                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>vlft/s</b>  | number of address translation page faults per second (valid page not present in memory)                                                                             |
| <b>pflt/s</b>  | number of page faults from protection errors per second (illegal access to page) or “copy-on-writes”, <b>pflt/s</b> generally consists entirely of “copy-on-writes” |
| <b>pgfil/s</b> | number of <b>vlft/s</b> per second satisfied by a page-in from the file-system (each <b>pgfil</b> causes two reads; see <b>sar -b</b> )                             |
| <b>rclm/s</b>  | number of valid pages per second that the system has reclaimed (added to list of free pages)                                                                        |

An example of **sar -p** output follows:

```

unix unix 3.2 2 i386 02/18/92
12:01:51 vlft/s pflt/s pgfil/s rclm/s
12:56:52 13.91 2.80 5.63 11.21

```

High values (over 100) for **vlft/s** can indicate that application programs are not efficient for a paging system (poor locality of reference), paging tunables need adjustment, or the memory configuration is inadequate for the system load.



The **vmstat(C)** command also reports statistics about paging activity. An example of the information provided by the **vmstat -s** command is:

```
17968 free swap space
12222 demand zero and demand fill pages
25932 pages on swap
44589 pages in cache
28719 pages on file
33791 protection fault
84644 pages are freed
 23 success in swapping out a process
 0 fail in swapping in a process
 22 success in swapping in a process
 98 swapping out a region
 64 swapping in a region
457461 cpu context switches
1870524 system calls
```

This output is representative of a system where excessive paging may be degrading performance.

### ***sar -q — process throughput***

---

The **sar -q** option reports the average queue length while the queue is occupied and percent of time occupied.

```
runq-sz run queue of processes ready to run
%runocc percentage of time the run queue is occupied
swpq-sz swap queue of processes to be swapped out
%swpocc percentage of time the swap queue is occupied
```

An example of **sar -q** follows:

```
unix unix 3.2 2 i386 02/18/92
11:00:56 runq-sz %runocc swpq-sz %swpocc
11:01:07 1.7 98 1.5 36
11:01:17 1.0 63 1.0 31
11:01:27 1.0 58 1.0 49
Average 1.3 74 1.2 39
```

If **%runocc** is greater than 90 and **runq-sz** is greater than 2, the CPU is heavily loaded and response time may be degraded. In this case, moving to a multiprocessing configuration might improve system response time. If **%swpocc** is greater than 20, more memory or fewer buffers would help reduce swapping/paging activity.

## **sar -r — page and swap usage**

---

The **-r** option records the number of memory pages and swap file disk blocks that are currently unused. The following are recorded.

**freemem** average number of 4K pages of memory available to user processes over the intervals sampled by the command

**freeswap** number of 512-byte disk blocks available for process swapping

An example of **sar -r** output follows:

```
unix unix 3.2 2 i386 02/18/92
12:01:51 freemem freeswap
12:56:52 208 5848
```

## **sar -u — CPU utilization**

---

The CPU utilization is listed by **sar -u** (or by the **sar** command with no options specified). At any given moment the processor is either busy or idle. When busy, the processor is in either user or system mode. When idle, the processor is waiting for input/output completion or has no work to do. The **-u** option of **sar** lists the percent of time that the processor is in system mode (%sys), user mode (%usr), waiting for input/output completion (%wio), and idle time (%idle). An example of **sar -u** follows:

```
unix unix 3.2 2 i386 02/18/92
09:20:08 %usr %sys %wio %idle
09:40:12 6 7 2 86
10:00:03 7 9 3 80
10:20:07 14 16 10 61
Average 9 11 5 76
```

In typical timesharing use, %sys and %usr are about the same value. In special applications, either of these can be larger than the other without anything being abnormal. A high %wio generally means a disk bottleneck. A high %idle, with degraded response time, may mean memory constraints; time spent waiting for memory is attributed to %idle.

The “idle” (percentage idle) column can also provide some insight into system performance. This figure is normally between 40 and 100 percent, even with a large number of active users. When this figure falls consistently below 30%, the chief competition for resources does not involve memory at all; the critical resource is raw processor power. (Run the **ps(C)** command to make certain that the excessive CPU usage is not due to a runaway process that is stealing every spare CPU cycle.)

If you are running a large number of users, it may help to switch to intelligent serial boards if you are using non-intelligent cards. Intelligent cards take some of the burden off the CPU rather than adding to the amount of work it has to do.

In addition, you should examine the files in */usr/spool/cron/crontabs* to see if jobs are queued up for peak periods that might better be run at times when the system is idle. Use the **ps** command to determine what processes are heavily loading the system. Encourage users to run large, non-interactive commands at off-peak hours. You may also want to run such commands with a low priority by using the **nice(C)** or **batch(C)** commands.

## ***sar -v — system tables***

---

The **-v** option reports the status of selected kernel tables. From this report you know when the size of these system tables need to be modified.

|         |                                                                                                                  |
|---------|------------------------------------------------------------------------------------------------------------------|
| proc-sz | number of process table entries presently being used/allocated in the kernel (controlled by <b>NPROC</b> )       |
| inod-sz | number of inode table entries presently being used/allocated in the kernel (controlled by <b>NINODE</b> )        |
| file-sz | number of file table entries presently being used/allocated in the kernel (controlled by <b>NFILE</b> )          |
| ov      | number of times an overflow occurred (one column for each of the above three items)                              |
| lock-sz | number of record lock table entries presently being used/allocated in the kernel (controlled by <b>FLCKREC</b> ) |

The values are given as level/table size, meaning the current number of table entries in use and the size of the table. An example of **sar -v** follows:

```

unix unix 3.2 2 i386 02/18/92

17:36:05 proc-sz ov inod-sz ov file-sz ov lock-sz
17:36:35 78/ 350 0 173/ 1050 0 153/ 1400 0 2/ 100
17:37:05 96/ 350 0 192/ 1050 0 169/ 1400 0 3/ 100
17:37:35 57/ 350 0 145/ 1050 0 146/ 1400 0 3/ 100

```

This example shows that all tables are large enough to have no overflows. Sizes could be reduced to save main memory space if these are the highest values ever recorded.

To see the actual number of these table entries being used currently, use the following command:

```
sar -v 1 1
```

The **pstat(C)** command provides similar information in a different format. To see the actual number of these table entries being used currently, you can also use this command:

```
pstat | grep '[0-9]'
```

## **sar -w — swapping activity**

---

The **-w** option reports swapping and context-switching activity. The following statistics are reported:

**swpin/s**    number of transfers into memory per second

**bswin/s**    number of 512-byte-block units (blocks) transferred for swap-ins (including initial loading of some programs) per second

**swpot/s**    number of transfers from memory to the disk swap area per second. If greater than 1, memory may need to be increased or buffers decreased.

**bswot/s**    number of blocks transferred for swap-outs per second

**pswch/s**    process switches per second. This should be 30 to 50 on a busy 4 to 6 user system.

An example of **sar -w** output follows:

```
unix unix 3.2 2 i386 02/18/92
19:53:44 swpin/s bswin/s swpot/s bswot/s pswch/s
19:53:58 0.0 0.0 0.0 0.0 37
19:54:14 0.0 0.0 0.0 0.0 39
19:54:24 0.0 0.0 0.0 0.0 39
Average 0.0 0.0 0.0 0.0 38
```

This example shows that there is sufficient memory for the currently active users, because no swapping is occurring. Many system administrators aim for zero swapping, which can be accomplished by adding physical memory. Disk accesses are significantly slower than memory accesses, so minimizing swapping activity is important for good performance.

The **vmstat(C)** command also provides information about swapping and context-switching activity. Use **vmstat -s** to view this data.

**sar -y — terminal I/O**

---

The **-y** option monitors terminal device activities. If you have a lot of terminal I/O, you can use this report to determine if there are any bad lines. Activities recorded are defined as follows:

**rawch/s** input characters (raw queue) per second

**canch/s** input characters processed by canon (canonical queue) per second. Canonical processing recognizes interrupt characters (**<Del>** by default), quit (**<Ctrl>\**), suspend (**<Ctrl>Z**), and editing characters such as **<Bksp>** and kill (**<Ctrl>U**). For example, if the user types:

**ha<Bksp>ello**

it increments the value of "rawch/s" by 7 and the value of "canch/s" by 5.

**outch/s** output characters (output queue) per second

**rcvin/s** receiver hardware interrupts per second

**xmtin/s** transmitter hardware interrupts per second

**mdmin/s** modem interrupts per second

Not all terminal drivers are written to produce these statistics. All SCO serial, console, and pseudo-tty drivers and most drivers produced by other vendors produce the rawch/s, canch/s, and outch/s statistics.

The number of modem interrupts per second (mdmin/s) should be close to 0. The ratio of xmtin/s to rawch/s and canch/s, and the ratio of rcvin/s to outch/s should be fairly constant. For non-intelligent I/O cards, these ratios should be close to 1:1. Intelligent I/O drivers move many characters per interrupt, and some drivers do not even use interrupts, so the ratios will be different. If the number of interrupts per transmitted character starts to increase dramatically, it usually indicates a bad line that is generating extraneous interrupts.

An example of **sar -y** output follows:

```

unix unix 3.2 2 i386 02/18/92
16:50:11 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
16:50:41 112 15 653 103 102 0
16:51:11 107 7 654 104 105 0
16:51:41 99 5 641 99 105 0
Average 106 9 649 102 104 0

```

## *swap(ADM) — check and add swap space*

---

Swap space is disk storage that is part of virtual memory. When main memory is exhausted, pages of processes that are sleeping awaiting their turn to execute can be written out to a swap device, thus freeing up main memory for executing processes. The system comes configured with one swap device. Adding additional swap devices with the **swap(ADM)** command makes more memory available to user processes. Excessive paging degrades system performance, but augmenting the swap space is a way to make more memory available to executing processes without tuning the kernel.

The following command adds a second swap device, */dev/swap1*, to the system. The swap area starts 0 blocks into the swap device and the swap device is 16000 512-byte blocks in size.

```
swap -a /dev/swap1 0 16000
```

Use the **swap -l** command to see statistics about all the swap devices currently configured on the system.

Running the **swap -a** command adds a second swap device only until the system is rebooted. To ensure that the second swap device is available every time the system is rebooted, use a startup script in the */etc/rc2.d* directory. For example you could call it *S09AddSwap*.

## *timex(ADM) — examine system activity per command*

---

The **timex** command times a command and reports the system activities that occurred on behalf of the command as it executed. Run without options, **timex** reports the amount of real (clock) time that expired while the command was executing and the amount of CPU time (user and system) that was devoted to the process. For example:

```
timex command command_options
real 6:54.30
user 53.98
sys 14.86
```

Running **timex -s** is roughly equivalent to running **sar -A**, but it displays system statistics only from when you issued the command until the command finished executing. If no other programs are running, this information can help identify which resources a specific command uses during its execution. System consumption can be collected for each application program and used for tuning the heavily loaded resources. Other information is available if the

process accounting software is installed; see the **timex(ADM)** manual page for information. To enable process accounting, remove the comment characters from the command lines in the */etc/rc.d/6/kinit* file and reboot the system.

## *tunesh(ADM) — automatic configuration tool*

---

**tunesh** is an automatic configuration utility supplied with SCO Open Desktop/SCO Open Server. It adjusts the values of key tunable parameters based on information you supply about your system. You should run **tunesh** soon after installing the system to determine initial values for these tunable parameters; as you add memory, terminals, filesystems, and so forth, you can run the configuration program again to modify system parameters accordingly.

**WARNING** **tunesh** sets system resources to levels appropriate for your system configuration, but may not result in optimum system performance. Study the **sar** reports and heed messages on the console about insufficient resources to understand the usage patterns for your particular system; in many cases, you may want to do additional tuning with the **configure(ADM)** command to allocate resources to meet your usage patterns.

**tunesh** bases the configuration on the information you supply about the following:

- amount of memory (RAM) installed (detected automatically)
- number of filesystems exported and imported via NFS
- number of local filesystems that will be mounted

On the server system, you are also asked about the following:

- number of X terminals
- number of serial login sessions (serial terminals and modems)
- number of remote login sessions (via **rlogin**, **telnet**, and **ftp**)

When running **tunesh** on the Personal System, the system makes calculations that assume 2 serial login sessions, 2 X-terminals, and 2 remote logins, to match the limits imposed by the Personal System license.

To reconfigure the system with **tunesh**:

1. Log in as **root** and issue the **/usr/lib/tunesh/tunesh** command.
2. Enter the new values as appropriate.
3. Relink the kernel, following the prompts.
4. When the program exits, shut the system down and reboot. If you are doing this during working hours, you should warn your users that the

system will be unavailable for 10-15 minutes. To shut down the system, make the following **sysadmsh** selection:

System ⇨ Terminate

If the system is under use, give people several minutes to finish their work and log out.

You can run the program again any time your configuration changes. Every time **tunesh** is used to change the system configuration, a mail message is sent to the *root* account with the heading "ODT System configuration log." This message contains a list of parameters changed and their new values, plus any error messages encountered during the tuning process. This mail serves as a record of the changes made to the system and should be filed with your System Log.

## ***XENIX system tools***

---

The **pstat(C)** and **vmstat(C)** commands that were originally developed for the XENIX operating system are also supported on SCO Open Desktop/SCO Open Server. These commands provide much of the same information available through the **sar(ADM)** and **crash(ADM)** commands, but in a different format. See the manual pages for more information.



## Chapter 4

# Kernel parameter reference

---

Kernel parameters control the allocation of kernel resources. These resources are constantly being used, released, and recycled, and include:

|                  |                                                                                                                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| buffers          | cache of in-memory storage units that hold recently used data. (Buffers increase efficiency by keeping this data on hand and decrease reading from the disk.) |
| table entries    | space in system tables that the kernel uses to keep track of current tasks, resources, and events                                                             |
| other parameters | other definable values that govern special resources (such as the number of multiscreens available or the quantity of semaphores)                             |

The use of these resources is defined by certain limits known as *tunable kernel parameters*. These limits can be decreased or extended, sometimes at the expense of other resources. Each resource or limit is represented by a separate kernel parameter. Deciding how to best optimize the use of these resources is known as performance or kernel tuning. Each time these tunable parameters are changed, their new values must be relinked (recompiled) into the kernel so that the new limits take effect.

Run the **tunesh(ADM)** utility after you install the operating system to select values that are appropriate for key parameters, based on answers about the configuration that you supply. These values generally provide a good starting point, but in most cases you can further improve the performance of your system by readjusting these values. There are several reasons for reallocating system resources:

- You install additional hardware memory and thus have greater memory resources to allocate. **tunesh** can be used to select new values for key parameters, but you may find that you can improve performance by adjusting certain parameters individually.

- Persistent error messages are displayed indicating that certain resources are used up, such as inodes or table entries.
- The system response time is consistently slow, indicating that other resources are too constrained for the system to operate efficiently (as when too little hardware memory is installed).
- Resource usage needs to be tailored to meet the needs of a particular application.

Before tuning your system, study the usage patterns for your system and decide which parameters need to be changed. “Understanding your performance goals” (page 869) provides suggestions to get you started. The best approach is to make small adjustments to the parameters, then study system statistics for a week or more and make additional adjustments. By making gradual adjustments, you reduce the chances of causing serious system problems.

If one of your performance goals is to reduce the size of the kernel (usually because the system is paging excessively), first concentrate on tunable parameters that control large structures. The following table summarizes the size ramifications for a select group of tunable parameters.

| Parameter   | Number of bytes                                |
|-------------|------------------------------------------------|
| NBUF        | 1096 (1024 for the buffer + 72 for the header) |
| NPBUF       | 72                                             |
| NHBUF       | 16                                             |
| NCLIST      | 12                                             |
| NINODE      | 76                                             |
| NHINODE     | 8                                              |
| NFILE       | 12                                             |
| NMPBUF      | 72                                             |
| NMPHEADBUF  | 72                                             |
| S5CACHEENTS | 72                                             |
| NPROC       | 344 (also influences value of NREGION)         |
| NREGION     | 68                                             |
| NBLKZ       | up to 5120                                     |

Specialized applications often require the reallocation of key system resources for optimum performance. For example, users with large databases may find that they need to lock more files simultaneously than the current allocation of file locks permit. Users who have no need for specialized features such as message handling may find that they can get a slight performance boost by deallocating those features.

Most of the tunable parameters discussed in this chapter are defined in the */etc/conf/cf.d/mtune* file. To manually change the values of specific tunable parameters, use the **configure**(ADM) command as discussed in “configure(ADM) — reallocate kernel resources” (page 882) or the **id tune**(ADM) command as discussed in “id tune(ADM) — reallocate kernel resources” (page 887).

**WARNING** Never edit */etc/conf/cf.d/mtune*. Whenever the system modifies a kernel parameter, an alternate entry is created in */etc/conf/cf.d/stune*, which overrides *mtune* with values specific to your system. The *mtune* file always contains the values these parameters had when the system was first installed, and can be a valuable reference.

This chapter lists the tunable parameters by the **configure** categories, in the order they are listed on the screens rather than alphabetically. The categories are:

1. Disks and buffers
2. Character buffers
3. Files, inodes and filesystems
4. Processes, memory management and swapping
5. Clock
6. MultiScreens
7. Message queues
8. Semaphores
9. Shared data
10. System name
11. STREAMS data
12. Event queues and devices
13. Hardware dependent parameters
14. Security
15. Asynchronous I/O parameters

The following categories of tunable parameters that are listed at the end of this chapter cannot be changed using **configure** (use **id tune**(ADM) instead):

- Boot load extension parameters
- Multiprocessing parameters
- Network-specific parameters
  - TCP/IP parameters
  - LAN Manager parameters

## Disks and buffers

---

**NDISK** number of disk drives attached to the system. This value is increased by `mkdev hd` if necessary when additional disks are installed. When the Diskinfo table overflows, the following message is displayed on the console:

CONFIG: dk\_name - Diskinfo table overflow (NDISK = *number* exceeded)

**NBUF** number of 1K system buffers allocated for the system buffer cache at boot time. The system buffer cache is a data cache or memory array used as a temporary storage area between the disk and user address space when reading to or writing from mounted filesystems.

If NBUF is set to 0, the system calculates the size of the buffer cache to a size that does not exceed the value of MAXBUF. `tunesh(ADM)` sets NBUF to 0 for configurations with more than 16 Megabytes of memory. For systems with less than 16 Megabytes of memory, `tunesh` assigns a smaller number of buffers. The size of the buffer cache is displayed as "kernel i/o/bufs" at boot time; see the `/usr/adm/messages` file. Cache hit rate increases with the number of buffers. Cache hits reduce the number of disk accesses and thus may improve overall performance. Study the `sar -b` report for statistics about the cache hit rate on your system. See "sar -b" (page 891) for more information.

The system buffer cache typically contains between 100 and 600 buffers, but may contain 8000 or more buffers on a large server system. Each buffer is fairly large, so having an unnecessarily large buffer cache can degrade system performance because too little space is available for executing processes.

For optimal performance, adjust number of hash queues (NHBUF) when you adjust the value of (NBUF).

**NPBUF** specifies how many physical I/O buffers to allocate. One I/O buffer is needed for each physical read or write that is active.

**NHBUF** specifies how many hash queues to allocate for 1K buffers. These are used to search for a buffer given a device number and block number rather than a linear search through the entire list of buffers. This value must be a power of 2. The NHBUF value should be chosen so that the value NBUF divided by NHBUF is approximately equal to 4.

- CTBUFSIZE** size of the tape buffer in kilobytes. This static buffer is allocated at `init` time and used by some tape devices such as the Q1C02 (*ct*) and SCSI (*stp*) tapes. This should have a value of 32 to 256. The following are reasonable values for associated circumstances:
- 32K bare minimum: insufficient to stream
  - 64K minimum to allow streaming (good for systems with little memory) or little tape use (performance is not critical)
  - 96K a first reduction, if default uses too much memory
  - 128K default: good tradeoff performance
  - 192K a first increase, if default provides poor performance
  - 256K maximum
- MAXBUF** maximum possible number of buffers in the buffer cache. This is the number of buffer description headers in the kernel. Fewer than this number of buffers may actually be autoconfigured by the kernel at boot time, depending on how much core is present. If **NBUF** is non-zero, then exactly **NBUF** buffers are configured, and there is no reason for **MAXBUF** to be larger than **NBUF**. If **NBUF** is 0, the kernel configures at most **MAXBUF** buffers automatically.
- DMAABLEBUF** number of transfer buffers for DMA requests over 16 megabytes. This must be a value of 4 to 128.
- PLOWBUFS** amount of buffer cache that is contained in the first 16 megabytes of RAM. It is expressed as a percentage and should be as high as possible, if you do not have a 32-bit controller (indicated by "`fts=d`" on the boot screen), to limit the number of copying requests made from buffers above 16 megabytes (see **NCOPYBUF**). The default value is 100%. This parameter needs to be changed only if you have more than 16 megabytes of RAM on your system.
- NCOPYBUF** number of buffers available for copying requests made from buffers above 16 megabytes to buffers below 16 megabytes, for non 32-bit controllers.
- NAUTOUP** specifies the buffer age in seconds for automatic filesystem updates. A system buffer is written to the hard disk when it has been memory-resident for the interval specified by the **NAUTOUP** parameter without a write operation, and the **bdflush** daemon process runs. So, if **NAUTOUP** is set to 10, a system buffer is written to the hard disk when **bdflush** runs after the

buffer has been memory resident for 10 seconds. Specifying a smaller limit increases system reliability by writing the buffers to disk more frequently and decreases system performance. Specifying a larger limit increases system performance at the expense of reliability.

- BDFLUSHR** specifies the rate in seconds for the **bdf flush** daemon process to run, checking the need to write the filesystem buffers to the disk. The range is 1 to 300. The value of this parameter must be chosen in conjunction with the value of **NAUTOUP**. For example, it is nonsensical to set **NAUTOUP** to 10 and **BDFLUSHR** to 100; some buffers would be marked dirty 10 seconds after they were written, but would not be written to disk for another 90 seconds. Choose the values for these two parameters considering how long a dirty buffer may have to wait to be written to disk and how much disk-writing activity will occur each time **bdf flush** becomes active. For example, if both **NAUTOUP** and **BDFLUSHR** are set to 40, buffers are 40-80 seconds old when written to disk and the system will sustain a large amount of disk-writing activity ever 40 seconds. If **NAUTOUP** is set to 10 and **BDFLUSHR** is set to 40, buffers are 10-50 seconds old when written to disk and the system sustains a large amount of disk-writing activity every 40 seconds. Setting **NAUTOUP** to 40 and **BDFLUSHR** to 10 means that buffers are 40-50 seconds old when written, but the system sustains a smaller amount of disk writing activity every 10 seconds. With this setting, however, the system may devote more overhead time to searching the block lists.
- PUTBUFSZ** specifies the size of a circular buffer, **putbuf**, that contains a copy of the last **PUTBUFSZ** characters written to the console by the operating system. The contents of **putbuf** can be viewed using **crash(ADM)**.
- PIOMAP** determines the size of the map entry array used by the kernel programmed I/O (PIO) breakup routine. This routine allows device drivers to do programmed I/O of large data blocks at interrupt level by breaking the data blocks into smaller data units. Users should not modify this parameter.
- PIOMAXSZ** maximum number of pages to use at one time for programmed I/O. Users should not modify this parameter.

## Character buffers

---

The following parameters control various data structure sizes and other limits in character device drivers provided with the operating system.

**NCLIST** specifies the number of character list buffers to allocate. Each buffer contains up to 64 bytes. The buffers are dynamically linked to form input and output queues for the terminal lines and other slow-speed devices. The average number of buffers needed per terminal is in the range of 5 to 10. Each entry (buffer space plus header) contains 72 bytes. When full, input and output characters dealing with terminals are lost, although echoing continues, and the following message is displayed on the console:

CONFIG: Out of clists (NCLIST = *number* exceeded)

**NEMAP** maximum number of I/O translation mappings. The value of this parameter should not be changed.

**NUMXT** number of layers a subdevice can configure to support bit-mapped display devices such as the BLIT or the AT&T 5620 and 730 terminals. When this number is exceeded, the following message is displayed on the console:

CONFIG: xtinit - Cannot allocate xt link buffers (NUMXT = *number* exceeded)

**NUMSXT** number of shell layers (shl(C)) a subdevice can configure.

**NKDVTTY** number of virtual terminals (ttys) supported by the console keyboard driver. Users should not modify this parameter.

**NCPYRIGHT** defines the size of a kernel data structure used to print vendor copyright messages when the system is booted. Users should not modify this parameter.

**MAX\_CFGSIZE** maximum size of configuration information saved by the **strings(M)** driver. This is the maximum size of information saved into the */dev/string/cfg* file described on the **string(HW)** manual page. If this limit is exceeded, the following message is displayed on the console:

CONFIG: string: Configuration buffer full (MAX\_CFGSIZE = *number* exceeded)

**PRFMAX** maximum number of text symbols that the kernel profiler (*/dev/prf*, can properly process. See **profiler(ADM)** for information about the kernel profiler.

**KDBSYMSIZE**

size of the kernel debugger symbol table. Must have a value of 50,000 to 500,000.

**TTHOG**

the size of the raw queue of the tty driver. The default value is 256. Increasing the value of this parameter allows more unprocessed characters to be retained in the tty buffer, which may prevent input characters from being lost if the system is extremely busy.

## *Files, inodes, and filesystems*

---

**NINODE**

specifies how many inode table entries to allocate. Each table entry represents an in-core inode that is an active file such as a current directory, an open file, or a mount point. Pipes, clone drivers, and XENIX semaphores and shared data also use inodes although they are not associated with a disk file. The file control structure is modified when changing this variable. The number of entries used depends on the number of opened files. The entries are normally in the range of 100 to 400, but could be much larger for large server systems. Each open file requires an inode entry in the inode table. If the inode table is too small, one of the following messages is displayed on the console:

CONFIG: ifreeget - Inode table overflow (NINODE = *number* exceeded)

CONFIG: s5iread - inode table overflow (NINODE = *number* exceeded)

When the inode table overflows, the specific request is refused. Although not fatal to the system, inode table overflow may damage the operation of various spoolers, daemons, the mailer, and other important utilities. Abnormal results and missing data files are a common result.

If the system consistently displays this error message, use `sar -v` command to evaluate whether your system needs tuning. The "inod-sz" column shows the number of inode table entries being used and the number of entries available in the table. The value for NINODE should be at least 10% greater than the number of active inodes at peak use time.

NINODE must be greater than or equal to NFILE.

**NHINODE**

specifies the size of the inode hash table; value must be a power of 2.



**NFILE** specifies how many open file table entries to allocate. Each entry represents an open file. The entry is normally in the range of 100 to 600, but could be much larger for large server systems. (NFILE must be less than or equal to NINODE.) The NFILE control structure operates in the same manner as the NINODE structure. When the file table overflows, the following warning message is displayed on the system console:

CONFIG: falloc - File table overflow (NFILE = *number* exceeded)

As a reminder, this parameter does not affect the number of open files per process (see the NOFILES parameter).

**NMOUNT** specifies how many mount table entries to allocate. Each entry represents a mounted filesystem. The *root* filesystem (/) is always the first entry. When full, the **mount(S)** system call returns the **EBUSY** error code. Because the mount table is searched linearly, this value should be as low as possible.

**CMASK** default mask used by **umask(S)** for file creation. By default, this is zero, meaning that the **umask** is not set in the kernel.

**ETRUNC** determines how the system handles attempts to use filenames that are longer than allowed for the particular filesystem architecture (either 14 or 255 characters). When ETRUNC is set to the default value of 0, attempts to create files with names greater than these respective limits fail with **ENAMETOOLONG**, which produces the *Filename too long* error message. This is the behavior mandated by POSIX FIPS requirements.

If ETRUNC is set to 1, filenames are silently truncated to either 14 or 255 characters depending on the filesystem architecture. Table 4-1 summarizes how the value of ETRUNC affects attempts to write filenames longer than supported on the filesystem:

**Table 4-1 Filename truncation**

| Filesystem | ETRUNC 0 | ETRUNC 1         |
|------------|----------|------------------|
| S51K       | Fails    | Truncates to 14  |
| XENIX      | Fails    | Truncates to 14  |
| AFS        | Fails    | Truncates to 14  |
| EAFS       | Fails    | Truncates to 255 |
| ES51K      | Fails    | Truncates to 255 |

**NOFILES** specifies the maximum number of open files per process. */bin/sh* uses three file table entries: standard input, standard output, and standard error (0, 1, and 2 are normally reserved for *stdin*, *stdout*, and *stderr*, respectively). This leaves the value of

**NOFILES** minus 3 as the number of other open files available per process. If a process requires up to three more than this number, then the standard files must be closed. This practice is not recommended and must be used with caution, if at all. If the configured value of **NOFILES** is greater than the maximum (11000) or less than the minimum (60), the configured value is set to the default (60), and a message is sent to the console. When the a process attempts to open more files than this limit, the following message is displayed on the terminal:

```
CONFIG: xsemfork - File table overflow (NOFILES = number exceeded)
```

Unless an application package recommends that **NOFILES** be changed, the default setting should be left as is.

- SHLBMAX** specifies the maximum number of shared libraries that can be attached to a process at one time. The maximum value is 16.
- FLCKREC** specifies the number of lock table entries, which determine the number of file regions that can be locked by the system. The "lock-sz" column on the `sar -v` report reports the number of entries being used in comparison to the number that are allocated. The proper value for this tuneable is determined by the applications being run on your system; if you are running applications that use file and record locking extensively, tune this to a higher value. If your applications use little file and record locking, the value of this tunable can be reduced.
- NMPBUF** number of Acer Fast File System cluster buffers. They are used to gather large disk requests before transferring the contents either to user space or to the buffer cache. `mpbufs` are large (16K each). The value **NMPBUF** should be 0 to 16, the latter being only appropriate only for a machine with 16 megabytes of memory or more and many users. When set to zero (the system default), the real value for this parameter is determined at boot time.
- NMPHEADBUF** number of Acer Fast Filesystem standalone cluster buffer headers. This parameter must have a value from 0 to 600.
- BFREEMIN** number of buffers that must be on the freelist before a buffer is waited for. It should usually be 0. Machines that are used primarily for media copying, `uucp` transfers, and other applications that are both quasi-single-user and access lots of files see a performance boost by setting this to a value of around `NBUF/10`. The maximum value is 100.

**S5CACHEENTS**

number of name components in the filename cache. Must have a value of 1 to 1024. The recommended value for diverse workgroups is to make S5CACHEENTS large, roughly three times the value of NINODE. See "Tuning disk I/O parameters" (page 872) for more information about choosing values for S5CACHEENTS, S5HASHQS, and S5OFBIAS.

**S5HASHQS** number of hash queues for name cache. Must have a value of 1 to 1021. The recommended value for diverse workgroups is to make S5HASHQS a prime number roughly a quarter the size of S5CACHEENTS.

**S5OFBIAS** determines bias towards keeping open files around in cache. Must have a value of 0 to 256. A value of zero means that the names of open files have no special caching priority. If S5OFBIAS is equal to S5CACHEENTS, the names of open files almost always remain in the cache.

**NGROUPS** maximum number of supplemental groups. This value has a default of 8 and a maximum of 16.

**HSNMOUNT** maximum number of mounted High Sierra (CD-ROM) filesystems.

**HSNINODE** maximum number of open inodes for High Sierra (CD-ROM) filesystems.

**HSNEXTENT** maximum number of file extents for High Sierra (CD-ROM) filesystems.

**DOSMOUNT** maximum number of mounted DOS filesystems.

**DOSINODE** maximum number of open inodes for DOS filesystems.

## *Processes, memory management, and swapping*

---

**NPROC** specifies how many process table entries to allocate. Each table entry represents an active process. The swapper is always the first entry, and */etc/init* is always the second entry. The number of entries depends on the number of terminal lines available and the number of processes spawned by each user. If the process table is filled up, users attempting to start new processes receive the following error message:

CONFIG: newproc - Process table overflow (NPROC = *number* exceeded)

The value of NPROC entry should be between 50 and 3,000. Use the *sar -v* report to see how many process table entries are being used compared to how many have been allocated. NPROC should be set to a value that is roughly 15% higher than the

largest value that shows in the *proc-sz* column of *sar -v*. Setting *NPROC* unnecessarily high wastes a lot of memory, especially since the proper value of the *NREGION* tunable parameter is determined relative to the value of *NPROC*.

- MAXUP** specifies how many concurrent user processes a non-super user is allowed to run. The entry is normally in the range of 15 to 60, with a maximum value of 300. This value should be at least 10% smaller than the value of *NPROC*. This value is per user identification number, not per terminal. For example, if 12 people are logged in on the same user identification, the default limit would be reached very quickly.
- NREGION** specifies how many region table entries to allocate. *NREGION* can be set between 100 and 10,000. As a general rule, set *NREGION* to a value that is a bit more than 3 times the value of *NPROC*. When the region table overflows, the following message is displayed on the console:
- CONFIG: allocreg - Region table overflow (*NREGION* = *number* exceeded)
- Most processes have three regions: text, data, and stack. Additional regions are needed for each shared memory segment and shared library (text and data) attached. However, the region table entry for the text of a "shared text" program is shared by all processes executing that program. Each shared-memory segment attached to one or more processes uses another region table entry.
- MAXPMEM** specifies the maximum number of physical pages. The default value of 0 specifies that all available physical memory be used.
- ULIMIT** specifies in 512-byte blocks the size of the largest file that an ordinary user can write. The default value is 2097151; that is, the largest file an ordinary user can write is one gigabyte. The super user can write a file as large as the filesystem can hold. The *ULIMIT* parameter does not apply to reads: any user can read a file of any size.
- SPTMAP** determines the size of the map entry array used for managing kernel virtual address space. Users should not modify this parameter.

The following six tunable parameters determine how often the *vhand* and *bdf* flush daemons run and under what conditions. *vhand* is the paging daemon that is responsible for freeing up memory as the need arises. It uses a "least recently used" algorithm to approximate process working sets, and it writes out those pages that were not modified during some period of time to the disk. The page size is 4096 bytes. When memory is exceptionally tight, the working sets of entire processes may be swapped out. The default values of these parameters should be adequate for most applications.

**AGEINTERVAL**

specifies the number of clock ticks a process runs before its pages are aged.

**GPGSLO**

specifies the low water mark of free memory in pages for **vhand** to start stealing pages from processes. Normally, **GPGSLO** is tuned to a value that is about 1/16 of pageable memory. Increase the value to make the daemon more active; decrease the value to make the daemon less active. This value must be a positive integer that is smaller than the value of **GPGSHI**.

**GPGSHI**

specifies the high-water mark of free memory in pages for **vhand** to stop stealing pages from processes. Normally **GPGSHI** is set to a value that is about 1/10 of pageable memory. Increase the value to make the daemon more active; decrease the value to make the daemon less active. The value must be a positive integer that is greater than the value of **GPGSLO**, and is usually not greater than 10 megabytes.

**GPGSMASK**

mask used by the paging daemon. The default is 0x00000420 (hex). Note that this appears as a decimal value in the system and should not be changed.

**MAXSC**

specifies the maximum number of pages that are swapped out in a single operation. The default value is 8 and should not be changed.

**MAXFC**

maximum number of pages that are added to the free list in a single operation. The default value is 1 and should not be changed.

**MAXUMEM**

maximum size of a user's virtual address space in 4096-byte pages. This value cannot be greater than 131072.

**MINARMEM**

minimum number of memory pages reserved for the text and data segments of user processes.

**MINASMEM**

threshold value that specifies the number of memory and swap pages reserved for system purposes (unavailable for the text and data segments of user processes). If this limit is exceeded, the following message is displayed on the console:

CONFIG: swapdel - Total swap area too small (MINASMEM = *number* exceeded)

**MINHIDUSTK**

specifies the minimum data relocation value such that the user stack and data can share a page table. The **MINHIDUSTK** and **MINUSTGAP** values should not be changed.

**MINUSTKGAP**

See **MINHIDUSTK**.

**MAXSLICE** specifies in clock ticks the maximum time slice for user processes. After a process executes for its allocated time slice, that process is suspended. The operating system then dispatches the highest priority process and allocates to it **MAXSLICE** clock ticks. **MAXSLICE** must be a value from 25 to 100.

## Clock

---

**NCALL** specifies how many call-out table entries to allocate. Each entry represents a function to be invoked at a later time by the clock handler portion of the kernel. This value must be in the range of 30 to 500. As a general rule of thumb, set **NCALL** to a value slightly higher than the number of enabled tty devices.

Software drivers may use call entries to check hardware device status. When the call-out table overflows, one of the following messages is displayed on the system console:

CONFIG: timeout - Timeout table overflow (NCALL = *number* exceeded)

CONFIG: MPX timeout - Timeout table overflow (NCALL = *number* exceeded)

It is difficult to determine exactly how many timeout table entries are required. Because each entry is relatively small, try doubling the value of **NCALL** after getting this message, which should be enough to solve the problem.

**TIMEZONE** specifies the **timezone** setting referred to in the **ctime(S)** system call. Note that the **timezone** value is a system default **timezone** and not the value of the **TZ** environment variable. This parameter can have a value from 0 to 1440.

**DSTFLAG** specifies the **dstflag** described for the **ctime(S)** system call. A value of 1 indicates Daylight Savings Time.

## MultiScreens

---

The multiscreen parameters determine the number of console multiscreens that can run simultaneously on the system. Each multiscreen requires about 6K of memory, so if you need to save memory and are not using multiscreens heavily, set **NSCRN** to 4 and **SCRNMEM** to 24. When you do this, you must also disable multiscreens 5-12 or **getty** generates warning messages when the system goes to multiuser state. **NSCRN** and **SCRNMEM** can be set to even smaller values if you are absolutely sure of the number of multiscreens that will be used.

**TBLNK** controls the console screen saver feature on VGA consoles. It is the number of seconds before the screen blanks to save wear on the monitor. **TBLNK** can have a value of 0 to 32,767, with zero disabling screen blanks. The default is 0.

|                |                                                                                                                                                                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NSCRN</b>   | number of multiscreens. A value of 0 configures this value at boot time based on the amount of memory installed. The maximum value is 12.                                                                                                                                                                                           |
| <b>NSPTTYS</b> | number of pseudo-ttys on the system. The maximum is 256. This parameter can be altered using the <b>mkdev pty</b> command, which also creates the additional device nodes. Pseudo-ttys are not related to console multiscreens; they are used for features such as serial multiscreens ( <b>mscreen(M)</b> ) and for remote logins. |
| <b>SCRNMEM</b> | number of 1024-byte blocks for console screen saves. A value of 0 configures this value at boot time based on the amount of memory installed. The maximum value is 128. Each multiscreen uses about 6K of memory, so when using a non-zero value for this parameter, make <b>SCRNMEM = 6*NSCRN</b> .                                |

## Message queues

---

The following tunable parameters are associated with interprocess communication messages:

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MSGMAP</b> | specifies the number of entries in the messages memory map. Entries in the message map table say that <i>n</i> <b>MSGSSZ</b> segments of memory are free at <i>x</i> address. This number really measures how fragmented you expect your map to get. It can be small if you always send a few large messages, and should be large if you send a lot of small messages. The safest number is roughly half of <b>MSGSEG</b> . The default value is 100. Each entry contains 8 bytes. |
| <b>MSGMAX</b> | maximum size of a message. The default value is 2,048. Although the maximum possible size the kernel can process is 64 kilobytes -1, the limit is 8,192.                                                                                                                                                                                                                                                                                                                           |
| <b>MSGMNB</b> | maximum number of bytes that all the messages in any one message queue can occupy. The default value is 4,096.                                                                                                                                                                                                                                                                                                                                                                     |
| <b>MSGMNI</b> | maximum number of message queues system-wide (id structure). The default value is 50.                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>MSGTQL</b> | number of messages that can be stored by the kernel; that is, the maximum number of unread messages at any given time. The default value is 40. Each entry contains 12 bytes.                                                                                                                                                                                                                                                                                                      |
| <b>MSGSSZ</b> | unit of memory allocation for storing a message. If <b>MSGSSZ</b> is 8, then <i>n</i> 8-byte segments of memory are used for storing a message. An 18-byte message would therefore require three message segments to be stored in a message queue. The default value is 8.                                                                                                                                                                                                         |

### MSGTQL MSGSEG

number of MSGSSZ segments of memory are allocated at kernel startup for holding messages. Therefore,  $\text{MSGSEG} \times \text{MSGSSZ}$  bytes of memory are allocated for messages. If MSGSEG is set at 0, then the kernel will auto-configure based on the total amount of system memory. If MSGSEG is 0, then MSGMAX and MSGMNB are both set to  $\text{MSGSEG} \times \text{MSGSSZ}$ .

**NOTE** The IPC\_NOWAIT flag can be passed into many of the `msg` system calls. If this flag is passed, then the system calls will fail immediately if there is no space for a message. If this flag is not passed, then the system calls will sleep until there is room for the message.

To determine adequate values for each of the parameters, compute the maximum size and number of messages desired, and allocate that amount of space. For example, if the system will have at most 80 messages of 1 kilobyte each pending, then MSGTQL should be set to 80, and MSGSEG is computed as:

- 80 messages of 1K each = 80 Kbytes total message space
- Divide total message space by MSGSSZ to get MSGSEG
- If  $\text{MSGSSZ}=8$ , then  $\text{MSGSEG}=80\text{K}/8=10\text{K}=10240$

## Semaphores

---

The following tunable parameters are associated with interprocess communication semaphores:

- |               |                                                                                                                                                                                         |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SEMAP</b>  | size of the control map used to manage semaphore sets. The default value is 10. Each entry contains 8 bytes.                                                                            |
| <b>SEMMNI</b> | number of semaphore identifiers in the kernel. This is the number of unique semaphore sets that can be active at any given time. The default value is 10. Each entry contains 32 bytes. |
| <b>SEMMNU</b> | number of undo structures in the system. The default value is 30. The size is equal to $8 \times (\text{SEMUME} + 2)$ bytes.                                                            |
| <b>SEMMSL</b> | maximum number of semaphores per semaphore identifier. The default value is 25.                                                                                                         |
| <b>SEMOPM</b> | maximum number of semaphore operations that can be executed per <code>semop</code> system call. The default value is 10. Each entry contains 8 bytes.                                   |
| <b>SEMUME</b> | maximum number of undo entries per undo structure. The default value is 10. The size is equal to $8 \times \text{SEMMNU}$ bytes.                                                        |



|                |                                                                                                                                                                                                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SEMVMX</b>  | maximum value a semaphore can have. The default value is 32,767, which is the maximum value for this parameter.                                                                                                                                                                                                     |
| <b>SEMAEM</b>  | adjustment on exit for maximum value, alias <b>semadj</b> . This value is used when a semaphore value becomes greater than or equal to the absolute value of <b>semop</b> , unless the program has set its own value. The default (and maximum) value is 16,384.                                                    |
| <b>SEMMNS</b>  | number of semaphores in the system. The default value is 60. Each entry contains 8 bytes.                                                                                                                                                                                                                           |
| <b>XSEMMAX</b> | size of the XENIX semaphore table that determines the maximum number of XENIX special semaphores allowed systemwide. The minimum value for <b>XSEMMAX</b> is 20, the maximum value is 60, and the default value is 60. When the XENIX semaphore table overflows, the following message is displayed on the console: |

CONFIG: xsem\_alloc - XENIX semaphore table overflow (XSEMMAX = *number* exceeded)

## Shared data

The following tunable parameters are associated with interprocess communication shared memory:

|                 |                                                                                                                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SHMMAX</b>   | maximum shared-memory segment size. The default value is 524,288.                                                                                                                                                                                                          |
| <b>SHMMIN</b>   | minimum shared-memory segment size. The default value is 1.                                                                                                                                                                                                                |
| <b>SHMMNI</b>   | maximum number of shared-memory identifiers systemwide. The default value is 100. Each entry contains 52 bytes.                                                                                                                                                            |
| <b>SHMSEG</b>   | number of attached shared-memory segments per process. The default value is 6. The maximum value is 15.                                                                                                                                                                    |
| <b>SHMALL</b>   | maximum number of in-use shared-memory text segments. The default value is 512.                                                                                                                                                                                            |
| <b>XSDSEGS</b>  | maximum number of XENIX special shared-data segments allowed system wide. The minimum value for <b>XSDSEGS</b> is 1, the maximum value is 150, and the default value is 25. When the XENIX shared data table overflows, the following message is displayed on the console: |
|                 | CONFIG: xsd_alloc - XENIX shared data table overflow (XSDSEGS = <i>number</i> exceeded)                                                                                                                                                                                    |
| <b>XSDSLOTS</b> | number of slots per XENIX shared data segment. The maximum number of XENIX special shared data segment attachments system wide is <b>XSDSEGS*XSDSLOTS</b> . The minimum value for <b>XSDSLOTS</b> is 1, the maximum value is 10, and the default value is 3.               |

## System name

---

**NODE** system name. The value of NODE must not be greater than eight characters.

## STREAMS data

---

STREAMS is a facility for UNIX system communication services. It supports the implementation of services ranging from complete networking protocol suites (such as TCP/IP) to individual device drivers. STREAMS defines standard interfaces for character I/O. The associated mechanism is simple and open-ended, consisting of a set of system calls, kernel resource and kernel routines.

STREAMS use system resources that are limited by values defined in kernel configuration modules. Depending on the demand that you and other system users place on these resources, your system could run out of STREAMS resources if you do not first reset the allocations in the kernel configuration modules. If you do not use STREAMS-based applications heavily and are trying to reduce the size of the kernel, reducing the values of these parameters (especially those that control the STREAMS data blocks) can free up a significant amount of memory.

Running out of some STREAMS resources (such as those controlled by the NQUEUE and NSTREAM parameters) generate kernel error messages. Running out of data blocks (controlled by the NBLK $\pi$  parameters) does not generate error messages, because the system will just use a larger buffer. However, running out of buffers of a certain size can cause performance problems, or may cause network processes and MultiScreens on UNIX System V and windows on SCO Open Desktop/SCO Open Server to terminate abnormally, or cause the system to lock or hang.

STREAMS parameters can be checked using the `strstat` function of the `crash(ADM)` utility. Before changing the STREAMS parameters NQUEUE, NSTREAM, NUMSP, or NBLK $\pi$ , check the `crash strstat` output. See “`crash strstat` (statistics about STREAMS buffers)” (page 884) for more information.

The following tunable parameters are associated with STREAMS processing.

**NQUEUE** number of STREAMS queues to be configured. Queues are always allocated in pairs, so this number should be even. A minimal STREAM contains four queues (two for the STREAM head, two for the driver). Each module pushed on a STREAM requires an additional two queues. A typical configuration value for systems that are not running X applications is

**4\*NSTREAM.** Some X applications use more than 8 queues for each NSTREAM pair, so a value of 5\*NSTREAM is more appropriate for systems that are running X clients. The following message is displayed on the console if you exceed the number of STREAMS queues:

CONFIG: id - out of STREAMS queues (NQUEUE = *number* exceeded)

**NSTREAM** number of STREAM-head (**stdata**) structures to be configured. One is needed for each STREAM opened, including both STREAMs currently open from user processes and STREAMs linked under multiplexers. The recommended configuration value is highly application-dependent, but a value of 32 to 40 usually suffices on a computer for running a single transport provider with moderate traffic. Each X client also uses a pair of **stdata** structures, so NSTREAM should be set to at least 100 on systems that are running X clients. When the number of STREAM head structures is exceeded, one of the following messages is displayed on the console:

CONFIG: stropen1 - Out of streams (NSTREAM = *number* exceeded)

CONFIG: stropen2 - Out of streams (NSTREAM = *number* exceeded)

**NBLK*n*** NBLK4 through NBLK4096 control the number of STREAMS data blocks and buffers to be allocated for each size class. Message block headers are also allocated based on these numbers: the number of message blocks is 1.25 times the total of all data block allocations. This provides a message block for each data block, plus some extras for duplicating messages (kernel functions **dupb()** and **dupmsg()**). The optimal configuration depends on both the amount of primary memory available and the intended application. When the system cannot allocate STREAMS blocks, the following message is displayed on the console:

CONFIG: sp: spclose - Cannot allocate STREAMS block (NBLK*nnnn* = *number* exceeded)

**NMUXLINK** maximum number of multiplexer links to be configured. One link structure is required for each active multiplexer link (STREAMS **L\_LINK** ioctl). This number is application-dependent; the default allocation of 87 guarantees availability of links.

**NSTRPUSH** maximum number of modules that may be pushed onto a STREAM. This prevents an errant user process from consuming all of the available queues on a single STREAM. By default this value is 9, but in practice, existing applications have pushed at most four modules on a STREAM.

**NSTREVENT** initial number of STREAM event cells to be configured. STREAM event cells are used for recording process-specific information in the **poll** system call. They are also used in the implementation

of the STREAMS `I_SETSIG ioctl` and in the kernel `bufcall()` mechanism. A rough minimum value to configure would be the expected number of processes to be simultaneously using `poll` times the expected number of STREAMS being polled per process, plus the expected number of processes expected to be using STREAMS concurrently. The default is 256. Note that this number is not necessarily a hard upper limit on the number of event cells that are available on the system (see `MAXSEPGCNT`).

#### **MAXSEPGCNT**

number of additional 4096-byte pages of memory that can be dynamically allocated for event cells. If this value is 0, only the allocation defined by `NSTREVENT` is available for use. If the value is not 0 and if the kernel runs out of event cells, it will under some circumstances attempt to allocate an extra page of memory from which new event cells can be created. `MAXSEPGCNT` places a limit on the number of pages that can be allocated for this purpose. Once a page is allocated for event cells, however, it cannot be recovered later for use elsewhere. The default value is 1.

**STRMSGSZ** maximum allowable size of the data portion of any STREAMS message. This should usually be set just large enough to accommodate the maximum packet size restrictions of the configured STREAMS modules. If it is larger than necessary, a single `write` or `putmsg` can consume an inordinate number of message blocks. The default value of 4,096 is sufficient for existing applications.

**STRCTLSZ** maximum allowable size of the control portion of any STREAMS message. The control portion of a `putmsg` message is not subject to the constraints of the minimum/maximum packet size, so the value entered here is the only way of providing a limit for the control part of a message. The default value of 1,024 is more than sufficient for existing applications.

**STRLOFRAC** percentage of data blocks of a given class at which low-priority block allocation requests are automatically failed. For example, if `STRLOFRAC` is 40 and there are forty-eight 256-byte blocks, a low-priority allocation request fails when more than nineteen 256-byte blocks are already allocated. The parameter helps prevent deadlock situations by starving out low-priority activity. The default value of 80 works well for most applications. `STRLOFRAC` must be greater than or equal to 0 and less than or equal to `STRMEDFRAC`. For slightly more efficient use of STREAMS buffers, increase the value of `STRLOFRAC` to around 95 and increase `STRMEDFRAC` as discussed next.

**STRMEDFRAC**

percentage cutoff at which medium priority block allocations are failed (see **STRLOFRAC**). The default value of 90 works well for most applications. For slightly more efficient use of **STREAMS** buffers, increase the value of **STRMEDFRAC** to around 98 and increase **STRLOFRAC** as discussed above. **STRMEDFRAC** must be greater than or equal to **STRLOFRAC** and less than or equal to 100. (There is no cutoff fraction for high-priority allocation requests; it is effectively 100.)

**NLOG**

number of minor devices to be configured for the log driver; the active minor devices are 0 through (**NLOG**–1). The recommended value of 3 services an error logger (**strerr**) and a trace command (**strace**), with one left over for miscellaneous usage. If only an error logger and a tracer are to be supported, this number can be set to 2. If there are several daemons for an application that may be submitting log messages, this number can be increased to accommodate the extra users.

**NUMSP**

determines the number of **STREAMS** pipe devices (*/dev/sp*) supported by the system. Users should not modify this parameter.

**NUMTIM**

maximum number of **STREAMS** modules that can be pushed by the Transport Layer Interface (TLI). This value controls the number of data structures used to hold pushed **STREAMS** modules configuration data. Users should not modify this parameter.

**NUMTRW**

number of Transport Layer Interface (TLI) read/write data structures to allocate in kernel data space. Users should not modify this parameter.

## Event queues and devices

---

**EVQUEUES**

maximum number of open event queues systemwide. The acceptable range is 1-16, with a default value of 8.

**EVDEVS**

size of the event table that determines the maximum number of devices attached to event queues systemwide. The acceptable range is 1-16, with a default value of 16. When the event table overflows, the following message is displayed on the console:

```
CONFIG: event - Event table full (EVDEVS = number exceeded)
```

##### EVDEVSPERQ

maximum number of devices per event queue. The acceptable range is 1-16, with a default value of 3. When the event channel overflows, the following message is displayed on the console:

```
CONFIG: event - Event channel full (EVDEVSPERQ = number exceeded)
```

## *Hardware dependent parameters*

---

**DMAEXCL** specifies whether simultaneous DMA requests are allowed. Some computers have DMA chips that malfunction when more than one allocated channel is used simultaneously. **DMAEXCL** is set to 0 by default to allow simultaneous DMA on multiple channels. If this causes a problem, set it to 1.

**KBTYPE** determines the logical character protocol used between the keyboard and the keyboard driver. This tunable is usually set to 0; a value of 1 specifies the AT logical character protocol which is recognized by the console driver but not by the X server. All AT-compatible keyboards support both modes.

##### VGA\_PLASMA

is set to 1 if a VGA gas plasma display is present, 0 if not.

**NSHINTR** maximum number of devices sharing the same interrupt vector. This has a default value of 8. Users should not modify this parameter.

**DO387CR3** controls the setting of high-order bits of Control Register 3 (CR3) when an 80387 math coprocessor is installed. Because of design defects in Intel's 80387 chip (B1 stepping,) the Intel 80387 math coprocessor may not operate correctly in some computers. The problem causes CPU to hang when DMA/paging/coprocessor accesses occur. You can work around this problem by changing the **DO387CR3** parameter from the default value of 0 to 1. The default value is 0 (switched off).

##### MODE\_SELECT

determines whether mode-select checking is performed on parallel ports. The default value of 1 enables mode-select checking. Setting this value to 0 disables this feature, and removes the printing delay incurred by these checks.

**WARNING** The following two parameters are not available via **configure(ADM)**. Use the **idtune(ADM)** command instead. Never edit the *mtune(F)* file.

- NAHACCB** number of mailboxes available for the Adaptec driver to talk to the Adaptec hardware. The higher the number, the less likely it is that the driver has to “sleep” before “talking” to the hardware. It is not normally necessary to modify this parameter.
- SDSKOUT** number of outstanding requests on a SCSI disk at any one time.

## Security

---

The security default (High, Improved, Low, or Traditional) can be selected as discussed in the chapter on administering user accounts in *Administering Operating System Services*; the security parameters can be set to modify the behavior of the security features and to ensure compatibility with utilities that expect traditional UNIX system behavior. Each of these parameters can be set to 0 (off) or 1 (on).

**SECLUID** controls the enforcement of LUID (login user ID). Under C2 requirements, every process must have a login user ID (LUID). This means that processes that set UIDs or GIDs, such as the printer scheduler (*lpsched*), must have an LUID set when started at system startup in */etc/rc2.d*. This can cause problems with *setuid* programs. When the security default is set to a lesser default (that is, not “High”), enforcement of login user ID (LUID) is relaxed and *setuid* programs do not require an LUID to run.

**SECSTOPIO** controls whether the kernel implements the *stopio(S)* system call. When **SECSTOPIO** is set to 1, the kernel acts on *stopio(S)* calls; when it is set to 0, the kernel ignores *stopio* calls. The *stopio* system call is used under C2 to ensure that a device is not held open by another process after it is reallocated. This means that other processes attempting to access the same device may be killed.

*stopio(S)* is used by *initcond(ADM)*, which is called by *getty(M)* immediately before starting user interaction and by *init(M)* immediately after an interactive session has terminated.

**SECCLEARID**

controls the clearing of SUID/SGID bits when a file is written. Under C2 requirements, the set user ID (SUID or *setuid*) and set group ID (SGID or *setgid*) bits on files must be cleared (removed) when a file is written. This prevents someone from replacing the contents of a *setuid* binary. This can cause problems with programs that do not expect this behavior. In the lower security defaults, SUID and SGID bits are not cleared when files are written.

The following table summarizes the initial settings of the security parameters for each security default.

| Parameter  | Low | Traditional | Improved | High |
|------------|-----|-------------|----------|------|
| SECLUID    | off | off         | off      | on   |
| SECSTOPIO  | off | on          | on       | on   |
| SECCLEARID | off | on          | on       | on   |

## Asynchronous I/O

The asynchronous I/O feature supports asynchronous I/O operations on raw disk partitions. It must be added to the kernel using the `mkdev aio` command for these parameters to have effect (see `aio(M)` for more information).

**NAIOPROC** size of the AIO process table that determines the number of processes that may be simultaneously performing asynchronous I/O. The default value is 5. When the AIO process table overflows, the following message is displayed on the console:

CONFIG: aio\_memlock - AIO process table overflow (NAIOPROC = *number* exceeded)

**NAIOREQ** size of the AIO request table that determines the maximum number of pending asynchronous I/O requests. The default value is 120. When the AIO request table overflows, the following message is displayed on the console:

CONFIG: aio\_breakup - AIO request table overflow (NAIOREQ = *number* exceeded)

**NAIOBUF** size of the AIO buffer table that determines number of asynchronous I/O buffers. This should always be set to the same value as **NAIOREQ**. When the AIO buffer table overflows, the following message is displayed on the console:

CONFIG: aio\_breakup - AIO buffer table overflow (NAIOBUF = *number* exceeded)

**NAIOHBUF** number of internal asynchronous hash queues.

**NAIOREQPP** maximum number of asynchronous I/O requests that a single process can have pending. The default value is 120, meaning that a single process can potentially exhaust all asynchronous I/O resources.



**NAIOLOCKTBL**

number of entries in the internal kernel table for asynchronous I/O lock permissions. The default value is 10. If there are many entries in the */usr/lib/aiomemlock* file, this value may need to be increased. When the AIO lock table overflows, the following message is displayed on the console:

CONFIG: aio\_setlockauth - AIO lock table overflow (NAIOLOCKTBL = *number* exceeded)

## Boot load extension parameters

---

**NOTE** This group of parameters is not available via `configure(ADM)`. Use the `idune(C)` command instead. Never edit the `mtune(F)` file.

**EXTRA\_NDEV**

number of extra device slots in `fmodsw[]`, `io_init[]`, and `io...[]`. It defines the number of slots reserved in the device driver tables for Boot Time Loadable Drivers.

**EXTRA\_NEVENT**

number of extra event slots. It defines the number of slots reserved in the event driver tables for Boot Time Loadable Drivers.

**EXTRA\_NFILSYS**

number of extra types of filesystem. It defines the number of extra types of filesystem that can be mounted at boot time.

**MAX\_BDEV** maximum number of block devices (`bdevcnt` is at least this value). It defines the minimum number of entries in `bdevsw[]`, the block device switch table.

**MAX\_CDEV** maximum number of character devices (`cdevcnt` is at least this value). It defines the minimum number of entries in `cdevsw[]`, the character device switch table.

## Multiprocessing parameters

---

**MAXACBUS** number of additional CPUs. Do not modify this parameter.

## ***Network-specific parameters***

---

Certain network-specific kernel parameters are not available through the **configure(ADM)** command, and some are not located in the *mtune(F)* file.

### ***TCP/IP parameters***

---

TCP/IP includes several variables located in *space.c* files in the */etc/conf/pack.d* directory. For a list of these parameters and their purpose, see “Kernel configuration” in the *Administering Networking Services* book.

### ***LAN Manager parameters***

---

LAN Manager adds several kernel parameters to the *mtune* file that are not available via the **configure** command. Instead, these are modified using the **id tune(ADM)** command. For a list of LAN Manager parameters and their purpose, see “Modifying kernel parameters: *mtune*” in the *Administering Networking Services* book.

## Chapter 5

# Troubleshooting reference

---

The rest of this book contains notes and tips about specific problems that may arise. These notes are organized alphabetically, grouped by related topic when appropriate (for example, security system troubleshooting and tape drive troubleshooting). Reading over this material leisurely before you have a problem may help you troubleshoot more quickly. If you are looking for information about a specific situation, you may find the index is the easiest way to access the specific piece of information you need.

## Boot troubleshooting

---

This section discusses reasons why a system that has booted successfully in the past may not boot now. These situations usually occur as a result of a power failure or system panic that corrupts the *root* filesystem, although configuration changes, hardware failure and human error can also cause these situations.

The *Installation and Upgrade Guide* gives instructions for troubleshooting a system that does not boot during installation.

If you cannot boot your system, first check the following:

- Is the system plugged in?
- Are any cables loose, disconnected, or misconnected?
- Are the cable chains terminated properly?
- Does the floppy drive contain a floppy that is not a boot floppy?
- Has your hard disk developed a bad track? A bad track on the disk may corrupt system files that are required for booting the system. See “Mapping a bad track” (page 942) for information on how to recover from this situation.

## *Restoring missing or corrupted system files*

---

On rare occasions, one or more of the critical system files is accidentally modified or removed, preventing the system from booting or operating correctly. In cases where your system does not boot, you must boot from floppy disks in order to access the system so that you can restore the critical files from backups.

To boot and access a system that does not boot from the hard disk, you must have made an Emergency Boot Floppy set as directed in the *Installation and Upgrade Guide*. This set consists of the boot floppy and the *root* filesystem floppy. The boot floppy contains three files necessary for booting and loading the UNIX system kernel: */boot*, */etc/default/boot*, and */unix*. The *root* filesystem floppy contains a subset of the UNIX system utilities that you can use to restore your system.

**NOTE** We recommend that you have a separate Emergency Boot Floppy set for each system or further corruption can result. Systems that have *identical* hardware and software configurations can share an Emergency Boot Floppy set.

If you have not made these floppies, you must reinstall the operating system. In some cases, if you do not also have a backup of the *root* filesystem, you must reinstall the operating system. To do this, follow the instructions in the *Installation and Upgrade Guide*.

The following sections discuss common reasons your system will not boot and give instructions for recovering. These conditions are listed in the order they will probably occur.

### */boot not found*

---

If your system displays the following message when you turn on the power to your computer, the */boot* file is missing:

```
/boot not found
Stage 1 boot failure: error loading /boot
```

The */boot* file contains the **boot(HW)** program, which loads and executes the kernel each time you turn on the computer.

If */boot* is missing, use the following procedure to boot the system from the Emergency Boot Floppy set so that you can then restore the file:

1. Insert the boot floppy in the drive and reboot the machine. This executes the initial boot from the boot floppy.

2. At the Boot: prompt, enter:

```
hd(40)unix
```

After you boot the system from the floppy, this command loads the kernel from the hard disk.

3. Bring up the system in single-user mode by entering the *root* password at the prompt.
4. Mount the boot floppy from the command line:

```
mount -r /dev/fd0 /mnt
```

The floppy must *not* be write protected.

5. While the floppy is in the drive, restore the */boot* file by entering the following command at the system prompt:

```
cp /mnt/boot /
```

This places a new copy of the */boot* file on the hard disk.

6. Before you remove the floppy disk from the drive, unmount the boot floppy (*/dev/fd0*) by entering:

```
umount /mnt
```

7. Remove the floppy disk from the drive and bring down the system using **haltsys**(ADM).
8. Reboot the system from the hard disk by pressing <Return> at the Boot: prompt.

## *unix not found*

If the system displays the “unix not found” message when you turn on the power to your computer, the */unix* file is missing. The */unix* file contains the bootable image of the UNIX system kernel. If */unix* is missing, you can boot from another kernel file, such as */unix.old*, */etc/conf/cf.d/unix*, or */etc/conf/cf.d/unix.old* by specifying the complete pathname of the file at the Boot: prompt.

If there are no other bootable kernel image files on the system, use the following procedure to boot the system from the Emergency Boot Floppy set so that you can restore the */unix* file:

1. Insert the boot floppy in the drive and reboot the machine.
2. At the Boot: prompt, enter:

```
fd(64)unix root=hd(40) swap=hd(41)
```

This loads the kernel from the boot floppy and mounts the *root* filesystem on the hard disk.

**NOTE** If you have changed the location of the swap device, you should substitute it for **hd(41)**.

3. Bring up the system in single-user mode.
4. Mount the boot floppy from the command line using:  

```
mount -r /dev/fd0 /mnt
```
5. While the floppy is in the drive, restore */unix* with the following command:  

```
cp /mnt/unix /
```

This copies the */unix* kernel file from the boot floppy to the hard disk.
6. Before you remove the floppy disk from the drive, unmount the floppy filesystem (*/dev/fd0*) by entering:  

```
umount /mnt
```
7. Remove the floppy from the drive and bring down the system with the **haltsys(ADM)** command.
8. Reboot the system by pressing **(Return)** at the **Boot:** prompt.
9. Relink the kernel to ensure that the correct devices are linked in.

### ***Cannot load floating point emulator***

---

If your computer does not have a 387 math coprocessor chip and the */etc/emulator* file is missing or corrupted, the boot fails with one of the following messages:

```
WARNING: Cannot load floating point emulator
WARNING: Zero length FP emulator file
WARNING: Cannot read emulator file header
```

(If the 387 chip is present, the kernel recognizes it in the hardware recognition boot message.)

If the boot fails with this message, use the following procedure to boot the system and restore */etc/emulator*:

1. Insert the boot floppy in the drive and reboot the machine.
2. At the **Boot:** prompt, press **(Return)** and when instructed, insert the *root* floppy. This boots the system and mounts the *root* filesystem from the floppies.
3. You should clean the *root* filesystem next:  

```
/bin/fsck -s /dev/hd0root
```

4. Working from the floppy disk, use the following **mount**(ADM) command to mount the hard disk *root* filesystem to */mnt*:

```
mount /dev/hd0root /mnt
```

If **mount** fails, refer to “Filesystem mount failed” (page 954) for information on checking the hard disk with **fsck**(ADM).

5. Copy */etc/emulator* from the *root* filesystem on the floppy to the mounted hard disk:

```
cp /etc/emulator /mnt/etc/emulator
```

6. Unmount the hard disk:

```
umount /mnt
```

7. Make sure that the floppy is still in the drive and reboot the system with the **halt**(ADM) command.
8. When you see the “Press any key” prompt, remove the floppy from the drive. Press **<Return>** at the Boot: prompt to boot from the hard disk.

## ***System hangs at login prompt***

---

If the system boots correctly but hangs at the login prompt after you enter multiuser mode, try other multiscreens or serial terminals.

**NOTE** It is also possible that the */etc/inittab* file has been corrupted. See “System cannot enter multiuser mode” (page 936) for more information.

If you cannot get a response from any ttys, the */bin/login* file may be missing. The */bin/login* file contains the **login**(M) program. This command is run at the beginning of each terminal session to allow users access to the system. Follow these steps to restore */bin/login*:

1. Power-cycle the machine and press **<Return>** at the Boot: prompt.
2. At the prompt, enter the *root* password to go into single-user mode.
3. Make the following selections from **custom**:

```
Install
SCO Open Server Enterprise System
Service Components
Operating System Services
Files
UNIX Run Time System
<F5>
login
<Return>
```

4. Exit **custom** when the file is restored.

## System cannot enter multiuser mode

---

If the system fails to enter multiuser mode when you press **(Ctrl)d**, or the following message is displayed at boot time, the */etc/inittab* file is missing:

```
INIT: Cannot open /etc/inittab errno: 2
INIT: SINGLE USER MODE
```

The */etc/inittab* file contains instructions for **init**. When *inittab* is missing, **init** cannot execute the system start-up instructions and the system cannot enter multiuser mode. When you press **(Ctrl)d**, the system remains in single-user mode and displays the error message above.

As */etc/inittab* contains instructions that reference other parts of the system, a special */etc/inittab* is written to the *root* filesystem floppy when you create the Emergency Boot Floppy set. Because of this, the procedure for restoring your system is more complex than simply copying */etc/inittab* from the floppy filesystem to the hard disk.

To restore *inittab*, you must recreate the kernel environment. To do this, use the following procedure:

1. Go into system maintenance mode.
2. Enter the following commands:

```
cd /etc/conf/cf.d
touch /etc.new_unix
../bin/idmkenv
```

3. You see the following:

```
The kernel environment includes device node files and /etc/inittab.
The new kernel may require changes to /etc/inittab or device nodes.
Do you want the kernel environment rebuilt? (y/n)
```

Enter **"y"** and press **(Return)**.

4. The following is displayed:

```
The kernel has been successfully linked and installed.
 To activate it, reboot your system.
Setting up kernel environment
```

5. Enter the **haltsys** command to reboot your system:

```
haltsys
```

When the system reboots, the new */etc/inittab* file is in place.



## *execvp of /bin/sulogin failed*

---

If the system displays the "INIT: execvp of /bin/sulogin failed; errno = 2" error message and goes directly into multiuser mode (run-level 2) at boot time, the */bin/sulogin* file is missing from the hard disk. The *sulogin*(ADM) utility must be present on the system to access single-user mode. If this file is missing, log in as *root* and make the following selections from *custom*:

```

Install
SCO Open Server Enterprise System
Service Components
Operating System Services
Files
UNIX Run Time System
<F5>
sulogin
<Return>

```

Exit *custom* when the file is restored.

## */etc/bcheckrc not found*

---

If the system displays the "INITSH: /etc/bcheckrc: not found" error message when you boot up, the */etc/bcheckrc* file is missing. The *init* utility executes *bcheckrc* according to instructions in the */etc/inittab* file whenever the system is booted. This utility checks the *root* filesystem and repairs it, if necessary. The */etc/bcheckrc* file should be on the hard disk when you boot the system.

If */etc/bcheckrc* is missing, use the following procedure to recover it:

1. Bring up the system in single-user mode.
2. Clean the *root* filesystem manually with *fsck* before doing anything on the system:

```
/etc/fsck -b /dev/root
```

3. When the filesystem has been checked and, if necessary, repaired, make the following selections from **custom**:

```
Install
SCO Open Server Enterprise System
Service Components
Operating System Services
Files
UNIX Run Time System
(F5)
bcheckrc
(Return)
```

Exit **custom** when the file is restored.

4. Enter **haltsys** at the prompt and reboot the system.

For more information on **bcheckrc**, see the **brc(ADM)** manual page.

### *System hangs at boot time*

---

If the boot process hangs after the “Kernel: i/o bufs” message, the */etc/init* file is missing from the system. The */etc/init* file contains the **init(M)** program. Once started, the **init** process spawns all other processes on the system, so if it is missing, no new processes are started.

Use the following procedure to restore */etc/init*:

1. Insert the boot floppy from the Emergency Boot Floppy Set in the floppy drive and reboot the machine.
2. Press (Return) at the Boot: prompt and when instructed, insert the *root* floppy.
3. Mount the hard disk *root* filesystem:

```
mount /dev/hd0root /mnt
```

If **mount** fails, check the hard disk with the **fsck(ADM)** command as discussed in “Filesystem mount failed” (page 954).

4. Copy the */etc/init* file from the *root* filesystem on the floppy to the mounted hard disk:

```
cp /etc/init /mnt/etc/init
```

5. Unmount the hard disk by entering **umount /mnt**.
6. With the floppy in the drive, reboot the system with the **haltsys(ADM)** command.
7. Remove the floppy from the drive when you see the “Hit any key” prompt. Press (Return) at the Boot: prompt to boot from the hard disk.

## Console troubleshooting

---

This section covers some common problems that you may have with the console on your system:

- The keyboard on your console locks up.
- How to prevent keyboard lockup by applying a special “patch”.
- The console does not recognize keyboard input because you selected the wrong console keyboard type.
- You cannot log into the console multiscreens when the system is in multi-user mode.

### Console keyboard locks up

---

When the system does not respond to input from the console keyboard, the situation is known as “keyboard lockup.” Console keyboard lockup only affects keyboards that are attached to the computer’s console, not standard terminals that are attached to serial lines.

You may be experiencing keyboard lockup if the following statements are true:

- The system console keyboard cannot be used to enter data or perform any tasks.
- You cannot use the <Ctrl><Alt><F1> through <F12> keys to switch multiscreens, and the <CapsLock> key does not turn the CapsLock light on or off.
- Other terminals on the system continue to work.
- Printers or other devices continue to work, and the system is still running.

Before trying to fix a locked keyboard, make sure that:

- You did not accidentally press <Ctrl>s (which stops the screen from scrolling). To check this, press <Ctrl>q and then see if you can enter characters from the keyboard.
- If your computer has a Keyboard Lock key, it is not in the locked position.
- The keyboard is plugged into the correct socket.
- The system itself is still running.

Check a terminal to see if it is still working and that you can perform system tasks, such as logging in and checking the date. If you do not have a terminal, watch the hard disk access light, if your computer has one.

**NOTE** You cannot use other terminals, and the hard disk access light may not flash if you are in single-user mode.

If it flashes periodically, at least once every 30 seconds, the system is still running and is using the hard disk.

If the console keyboard is still locked after checking these suggestions, you can try unplugging the console keyboard and plugging it in again.

**WARNING** Unplugging the keyboard and reconnecting it while the system is powered up can damage some computers.

If this fixes the problem, your situation is definitely keyboard lockup. If this last step does not fix the problem, you may still have keyboard lockup.

### *Preventing console keyboard lockup*

---

You can prevent keyboard lockup by applying a special “patch” that changes the operating system kernel.

**NOTE** This patch disables the keyboard lights, so you should use it only if you have tried the other approaches.

The procedure for using the patch is as follows:

1. Get the system console working, if it is not. Reboot the system if you have to and bring it up in single-user mode.

If you did not reboot, log in as *root* on the system console and shut the system down to single-user mode with the **shutdown** command:

```
/etc/shutdown su
```

(For more information, see the **shutdown(ADM)** manual page.)

2. Once the system is in single-user mode, back up the kernel with the following commands:

```
cd /
cp unix unix.00
```

3. Now, patch the kernel with the following commands:

```
/etc/_fst -w /unix
ledspresent/w 0
$q
```

4. Shut down the system using **/etc/shutdown**.
5. When you see the “Normal System Shutdown” message, press any key to reboot the system. You have now fixed the keyboard lockup problem.

You can apply this patch permanently by entering the following commands:

```
cd /etc/conf/pack.d/cn
copy -om Driver.o Driver.o.save
/etc/_fst -w Driver.o
ledspresent/w 0
$g
```

## *Cannot log into console*

---

If you try to log into the console in multiuser mode, and the system displays the following error message:

```
Cannot obtain database information on this terminal
```

Refer to “Cannot obtain database information on this terminal” (page 978) for more information.

## *cron, at, and batch troubleshooting*

---

**cron(C)** is the clock daemon that schedules jobs to run automatically at a set time. The **at** and **batch** commands can be used by users to run non-interactive jobs once. See the **at(C)** manual page for more information.

### *cron daemon is not running*

---

The **cron** daemon executes commands submitted with the **at(C)**, **batch(C)**, and **crontab(C)** commands at specified dates and times. (The **cron** daemon is started automatically by a script in the **/etc/rc2.d** directory when the system is started in multiuser mode.) If **cron** is not running and you try to submit a job with one of these programs, the system displays the following message:

```
cron may not be running - call your system administrator
```

Another indicator that **cron** is not running is if previously scheduled jobs are not being executed. Use the following command to see if the **cron** daemon is running:

```
ps -ef | grep cron
```

If there is no **cron** process, then the daemon is not running. To start **cron**, log in as **root** and enter the following command:

```
sd cron
```

If this does not start the **cron** process, simply reboot the system. During startup, the system displays a message like the following:

```
! *** cron started *** pid = 140 Tue Apr 21 14:02:47 PDT 1992
```

The jobs scheduled with **at**, **batch**, and **crontab** should now execute properly. If **at** and **batch** continue to report error messages, refer to “at command fails: cannot change mode of job” (page 942) for more information.

**NOTE** If this is a persistent problem, you can start **cron** logging by entering **CRONLOG=YES** in the **/etc/default/cron** file. After **cron** is started, the **/usr/spool/cron/log** file will hold a record of all **cron** transactions; this information may help you determine why **cron** is halting.

## *at command fails: cannot change mode of job*

---

If you do not have the kernel authorization, **chmodsugid**, the system displays the following message when you try to invoke **at(C)** or **batch(C)**:

can't change mode of job

To grant **chmodsugid** authorization so that the user can use **at**, use the following procedure:

1. Log in as *root*.
2. Invoke **sysadmsh** and select:  
Accounts ⇨ User ⇨ Examine:Privileges
3. Add "chmodsugid" to the Kernel Authorization field.

To allow the changes to take effect, the user must log out and then log in again.

This is different from the situation in which the user is not *authorized* to use **at**. If a user who is not authorized tries to use **at**, the system displays:

at: you are not authorized to use at. Sorry.

To allow a user to use **at**, use the **sysadmsh(ADM)** selection Jobs ⇨ Authorize ⇨ Scheduled ⇨ User.

## *Disk problems*

---

This section covers some common problems that you may have with the hard disk on your system:

- bad tracks
- disk load balancing

### *Mapping a bad track*

---

The hard disk sometimes develops a bad track, which is an area of the disk that is not reliable for data storage. The hard disks supported on SCO systems allow you to map the bad track into the bad track table that lists any areas of the hard disk that should not be used. The bad tracks listed on the table are aliased to good tracks so that the operating system avoids the areas of the disk that cannot be read or written.

If your hard disk develops a bad track after the system is installed and running, an error message like the following is displayed on the console:

```
wd: ERROR on fixed disk ctrlr=0 dev=0/47 block=31434 cmd=00000020
 status=00005180, sector = 62899, cylinder/head = 483/4
```

If this error occurs, use **badtrk**(ADM) to create a new bad track table so that the system can avoid the new bad track(s). The **badtrk** utility is a menu-driven utility for viewing, adding, or deleting entries in the bad track table.

**WARNING** SCSI disks do not use **badtrk**. Use the **scsibadblk**(ADM) utility to map any bad blocks. If your adapter supports Automatic Write Remapping (AWRE), you can use **scsibadblk** to enable this feature.

To use **badtrk**, do the following:

1. Reboot the system and enter single-user mode. Do not use **shutdown su**.
2. Enter **badtrk** at the system prompt.
3. Select option 2 to scan the disk, then select option 1 to scan the entire UNIX system partition.
4. Indicate whether you want to do a quick or thorough scan and, at the destructive scan prompt, enter "n".

**NOTE** Run **badtrk** in nondestructive mode to save the data on your hard disk. The thorough scan is recommended if new bad tracks have appeared.

5. When the scan is complete, the main menu reappears. Select "q" to return to the system.

The **badtrk** utility automatically enters any flaws it detects in the bad track table (as long as there is room in the table). For more information on the options to **badtrk**, see the manual page on **badtrk**(ADM).

## Disk load balancing

If your system is configured with more than one disk, it is important to spread the activity over all disks fairly equally. Use the **sar -d** command to get statistics about disk usage. The **sar**(ADM) manual page describes all the columns on the **sar -d** report; it is usually most interesting to look at the "Average" information printed at the bottom of the report. Numbers that are radically different between disk devices (especially in the "%busy", "avwait", and "avserv" columns) may indicate that file systems should be rearranged between your disks.

Note that `sar -d` reports on the disk activity occurring at the moment the data collector ran. If the data collector is running every 15 or 30 minutes and the disk load varies, individual numbers may be misleading. Nonetheless, the averages over a period of time can be very useful in evaluating whether your filesystems and swap device(s) are arranged to balance disk activity between all disks. See “`sar -d`” (page 893) for more information.

Specific steps to take in balancing data across all disks include:

- If you are running a large database application, put the database on a separate disk from the one that holds the *root* filesystem.
- Use the `swap(ADM)` command with the `-a` option to add swap devices to other disks. If your swapping activity is heavy, put additional swap devices on disks that do not contain other heavily-used files.

## Filesystem troubleshooting

---

UNIX System V uses filesystems to store files on disk. Each filesystem corresponds to a disk division, which is identified as a file in the */dev* directory. Many system error messages identify a filesystem by the */dev* name rather than by the filesystem name that appears, for example, as output from the `df(C)` command. See the filesystems chapter in *Administering Operating System Services* for more information about administering filesystems; the sections here discuss common problems that may arise with filesystems.

### Out of inodes on filesystem

---

An inode is an internal structure that the operating system uses to track control information about a file such as size and last access date. Each file uses one inode. When a filesystem runs out of inodes, the system displays the following error message:

```
NOTICE: s5ialloc - Out of inodes on EAFS dev hd (nn/mm)
```

where *nn/mm* is the filesystem that has run out of free inodes. To fix this problem:

1. Remove unnecessary (old, temporary, *core*, or log files) files from the filesystem.
2. Use the `find(C)` command with the `-size` parameter to see if the filesystem contains a large number of small files. The initial allocation of inodes assumes a ratio of about four data blocks per inode. If the filesystem contains mostly files that are smaller than four blocks, it runs out of inodes.



The number of inodes available on a filesystem is determined when the filesystem is created (using `mkfs(ADM)`). If the filesystem consistently runs out of free inodes, you can reconfigure the filesystem and increase the number of inodes. To do this:

1. Back up the filesystem using the `sysadmsh` selection:  
Backups ⇨ Create ⇨ Unscheduled
2. Verify the integrity of the backup using the `sysadmsh` selection:  
Backups ⇨ Integrity
3. Unmount the filesystem using the `sysadmsh` selection:  
Filesystems ⇨ Unmount
4. Run `mkfs` from the command line and specify more inodes for the filesystem.

**NOTE** This procedure destroys the information on your hard disk. Do not use the `mkfs` command without a complete and verified backup.

For example, to reconfigure the number of inodes on the `/dev/u` filesystem to 6400, use the following `mkfs` command:

```
mkfs /dev/u fssize:6400
```

*fssize* must be replaced with the size of the filesystem in question, in 1K blocks. This value can be obtained as described in “Repairing a filesystem when `fsck` stops at size check” (page 951). For more information on the `mkfs` command, see the `mkfs(ADM)` manual page.

5. Mount the filesystem using the `sysadmsh` selection:  
Filesystems ⇨ Mount
6. Restore the filesystem from the backup with the following `sysadmsh` selection:  
Backups ⇨ Restore ⇨ Full

## Out of space on filesystem

File data is stored in data blocks in the filesystem. Use the `df(C)` command to see the number of data blocks and inodes that are available. When a filesystem has little or no space left to work, the system displays the following message:

```
NOTICE: clalloc - Out of space on EAFS dev hd (nn/mm)
```

When the filesystem runs out of space, the system stops any attempts to write to the filesystem. The only way to restore system operation is to delete or reduce files from the named filesystem.

Use the following suggestions to restore space to the filesystem:

1. Use the **wall(ADM)** command to send a system-wide message asking users to remove unnecessary files.
2. Check for the number of blocks used by each file and directory in the named filesystem with the **du(C)** command. For example, the following command sequence displays the number of blocks in the directories under the **/u** filesystem:

```
cd /u; du
```

Within a directory, the following command sorts the files by size and displays them with the largest first:

```
ls -lsa | sort -nr | pg
```

You can then send mail, asking the users who own the largest files to remove or compress them.

3. Use the **find(C)** command to locate exceptionally large or old directories and files, and send mail to the owners asking them to remove unnecessary files.
4. Use **find** to locate and remove temporary files, *core* files and any unused *a.out* files. Remove files from the */usr/preserve* and */lost+found* directories. In addition, you can modify the */etc/default/cleantmp* file and associated **crontab** entry that tells **cleantmp(ADM)** how often to clear key directories (*/tmp* and */usr/tmp* by default) of files.
5. Clear the contents of system log files, such as */usr/spool/lp/log*, */usr/adm/messages* and */usr/adm/sulog* (if enabled), and any log files for add-on programs. If disk space is tight, you can also delete data under the */usr/adm* directory, such as the **sar** data in the */usr/adm/sa* directory or the accounting data (if enabled) in the */usr/adm/acct* directory. This material can be first archived to tape or floppy if you think you may need to access it again.

To clear a log file, use the following construction:

Bourne or Korn shell:

```
> filename
```

C shell:

```
cat /dev/null > filename
```

6. If you have security auditing enabled, check the disk usage in the */tcb/audittmp/audit\** directories. Back up and remove old audit files using the **sysadmsh** selections:

```
System ⇄ Audit ⇄ Files ⇄ Backup
```

```
System ⇄ Audit ⇄ Files ⇄ Delete
```

You should check and remove these files regularly.

7. Refer to “Reducing disk fragmentation” (this page) for the procedure to reduce disk fragmentation.
8. If the system is chronically short of free space, add additional hard disks and/or create and mount additional filesystems.

Each of these steps is described in more detail in the section on maintaining free space in filesystems in the chapter on managing filesystems the filesystems chapter in *Administering Operating System Services*

## Checking free space on filesystems

---

You should periodically check the amount of free space in your filesystems with the `sysadmsh` selection:

System ⇄ Report ⇄ Disk

This command prints the amount of space left in the filesystem in 512-byte blocks. If the space is low, see the previous section for hints on what to do (before you run out of space).

## Reducing disk fragmentation

---

If your system has been in use for some time, the constant creation and removal of files creates a situation called *disk fragmentation*. This means that the files in the filesystem are written in small pieces scattered widely across the hard disk. This results in increasingly poor disk I/O performance. Fragmentation may become a problem when the disk is (approximately) more than 75% full.

**NOTE** The following procedure cannot be used on the *root* filesystem.

To reduce disk fragmentation, first make a complete backup of all the files in the filesystem:

1. Invoke `sysadmsh` and select:  
Backups ⇄ Create ⇄ Unscheduled
2. Select the filesystem and media device (block size is selected automatically) and format floppy disks, if necessary.
3. Load a volume, tape or disk, into the selected drive, and press ⟨Return⟩. The system copies files to the drive.
4. Next, verify the integrity of your backup. Invoke `sysadmsh` and select:  
Backups ⇄ Integrity  
Enter the media type and insert each volume of the backup in turn.

5. Use the **mkfs(ADM)** or **divvy(ADM)** command to recreate the filesystem.
6. Run **fsck(ADM)** to verify the integrity of the filesystem, then mount the filesystem:

```
mount /dev/u /u
```

7. Restore the files from the backup. Invoke **sysadmsh** and select:

```
Backups ⇨ Restore ⇨ Full
```

Insert the first volume and enter the name of the filesystem that you want to restore and the media device. The actual **cpio** command line is displayed.

Because the files are completely rewritten on the disk, each file is written in one piece and fragmentation is reduced. It is a good idea to reduce disk fragmentation about once a year on a heavily-used system and less often on a lightly-used system. Verify that you have complete, accurate, and readable backups before you begin or you will lose files.

### *Restoring a corrupted root filesystem*

---

If your *root* filesystem is so corrupted that, when you boot the system, **fsck(ADM)** cannot run, use the following procedure to restore your system:

1. Insert the boot floppy in the drive and reboot the machine.
2. At the Boot: prompt, press <Return>. At the prompt, insert the *root* filesystem floppy. This boots the system and mounts the *root* filesystem from the floppies.
3. At the system prompt, enter:

```
/etc/fsck /dev/hd0root
```

You should see messages indicating that **fsck** is proceeding through five or six phases of system cleaning. If **fsck** exits within a few seconds or the system displays error messages that make no sense, you must restore the entire *root* filesystem from backups. Here are two examples of nonsense messages:

```
UNKNOWN FILE SYSTEM VERSION 65535
CLEANING NON SYSTEM 3 FILESYSTEM
```

If **fsck** appears to be successful, shut down the system with **haltsys(ADM)** and boot from the hard disk by pressing <Return> at the Boot: prompt.

If **fsck** is not successful, you must recreate the filesystem and restore it from backups. See “Restoring your root filesystem from backups” (page 949) for more information.

If you cannot boot from your hard disk at this point, you may have to reinstall the operating system from scratch. See the *Installation and Upgrade Guide* for more information.

## Restoring your root filesystem from backups

To restore your *root* filesystem, you must have the following:

- Either a *root* filesystem backup created with `sysadmsh(ADM)` (either scheduled or unscheduled) or a full filesystem backup created with `cpio(C)`.
- The Emergency Boot Floppy Set that you created immediately after installing your system.

**NOTE** The Emergency Boot Floppy Set must have a kernel with the correct tape drivers installed. If the tape drivers are not present in the kernel on the Boot floppy, or if it is not configured with the correct parameters, the restore process will fail.

If you do not have an Emergency Boot Floppy Set, use the procedure described in “Restoring a root filesystem without an Emergency Boot Floppy Set” (page 950) to restore your *root* filesystem.

### Restoring a root filesystem with an Emergency Boot Floppy Set

Use the following steps to restore your *root* filesystem:

1. If the system is not already down, bring it down. See the chapter on starting and stopping the system in *Administering Operating System Services*.
2. At the prompt to reboot, insert your N1 (Boot) floppy disk in the drive and press (Return).
3. At the Boot: prompt, enter `defbootstr`.

**NOTE** If `defbootstr` results in an error, enter `restart`.

4. When prompted, insert the N2 (Filesystem) disk and press (Return).
5. Continue with the installation procedure to reinitialize your hard disk and recreate the filesystems.

For more information, see your *Installation and Upgrade Guide*.

**NOTE** If the hard disk or filesystem information is still valid, you do not need to recreate it. For example, if your hard disk still has a valid `badtrk` table, you do not need to run a `badtrk` scan.

After you see the “Making Filesystems” message, you are prompted to insert the N1 disk at the Boot: prompt.

6. Insert the Boot floppy from your Emergency Boot Floppy Set and press (Return). Do not use the N1 floppy.
7. At the prompt, insert your Root floppy from your Emergency Boot Floppy Set and press (Return). Do not use the N2 floppy.

8. Enter:

```
fsck /dev/hd0root
```

This cleans the *root* filesystem on the hard drive.

9. Enter:

```
mount /dev/hd0root /mnt
```

This mounts the *root* filesystem on the hard disk.

**NOTE** If you see the following message, you can safely ignore it:

```
mount: warning /dev/hd0root /mnt already in /etc/mnttab
```

10. Restore the filesystem with the following commands:

```
cd /mnt
```

```
cpio -ivmkBud -I/dev/xxx
```

where *xxx* is either *rct0* (for a 1/4 inch cartridge tape) or *rctmini* (for a mini-cartridge tape).

11. Unmount and clean the newly restored filesystem with the following commands:

```
cd /
```

```
umount /dev/hd0root
```

```
fsck /dev/hd0root
```

12. Shut down the system by entering */etc/shutdown*.

13. At the prompt to reboot, remove the floppy from the floppy drive and press *<Return>* to boot from the hard disk.

14. Restore any secondary filesystems by entering *sysadmsh* and making the following selection:

Backup ⇌ Restore

### Restoring a root filesystem without an Emergency Boot Floppy Set

Use the following procedure to restore your *root* filesystem if you did not create an Emergency Boot Floppy Set at installation time. (You must install the Runtime System before you perform the restore.)

1. Perform steps 1 through 5 in “Restoring a root filesystem with an Emergency Boot Floppy Set” (page 949).
2. At the Boot: prompt, insert the N1 floppy and press *<Return>*.
3. Continue with the installation procedure to install the complete Runtime System. See your *Installation and Upgrade Guide* for details.
4. At the prompt to reboot, remove the floppy from the floppy drive and press *<Return>* to boot from the hard disk.

5. At the “INIT: SINGLE USER MODE” message, enter the *root* password to enter system maintenance (single-user) mode.
6. Enter **mkdev tape** to install the tape drive. (See the chapter on adding tape drives in the *Hardware Configuration Guide* for more information.)
7. At the following prompt, enter “y”:

The link kit is only partially installed. Do you wish to install it?

8. Insert the installation floppy disks when prompted.
9. Continue with the tape drive installation procedure and relink the kernel.
10. Enter **reboot** to reboot the system.
11. Verify that you can write and read from the tape drive.

For example, insert a tape in the drive and enter:

```
tar cv8 /etc/motd
tar xv8 /etc/motd
```

12. Create the Emergency Boot Floppy Set using the following **sysadmsh(ADM)** selection:

Filesystems ⇔ Floppy

For more information, see the chapter on creating an Emergency Boot Floppy Set in your *Installation and Upgrade Guide*.

13. Shut down the system. See the chapter on starting and stopping the system in *Administering Operating System Services*.
14. At the prompt to reboot, insert the Boot floppy from your Emergency Boot Floppy Set and press (Return).
15. At the prompt, insert your Root floppy from your Emergency Boot Floppy Set and press (Return).
16. Follow steps 8 through 14 in “Restoring a root filesystem with an Emergency Boot Floppy Set” (page 949) to restore your *root* filesystem (and any secondary filesystems) from backups.

## Repairing a filesystem when fsck stops at size check

If the super block for a filesystem is so badly damaged after an abnormal shut-down or hardware failure that **fsck(ADM)** quits at the size check or reports an unusually large number of errors, you can use the **fsdb(ADM)** (“filesystem debugger”) utility to patch the super block by hand.

The **fsck** utility reads the size of the filesystem from the super block. If the values currently stored in the super block for **FSIZE** (the total number of blocks in the filesystem) and **ISIZE** (the number of inodes allocated in the filesystem) are not the normal values for the filesystem, **fsck** displays the results of the size check and quits, as in the following example:

```
fsck /dev/root
/dev/root
/dev/root File System: / Volume: root
Size check: FSIZE 0 ISIZE 0
#
```

When using **fsdb**, you have to change only one of the two values in the super block, **FSIZE** or **ISIZE**, back to their normal values for your filesystem.

**NOTE** You cannot run **fsdb** on the *root* filesystem if you are unable to boot up the system. If you have **fsdb** on the *root* floppy, you can use that version to repair the *root* filesystem.

Because **fsdb** is a powerful tool that allows you to directly change the super block (which contains important information about your filesystem), be very careful when using this tool. If you enter improper values with **fsdb**, you can permanently damage your filesystem and lose all filesystem data. The **fsdb(ADM)** manual page describes a number of other ways in which you can use **fsdb** to look at and manipulate the super block. The use of such advanced features is recommended only for the experienced system administrator.

**NOTE** Make sure that the filesystem is unmounted before using **fsdb** to repair it.

To repair the damage described above, follow this procedure:

1. Enter the following command:

**divvy**

**NOTE** When **divvy** is invoked without an argument, it uses the active partition on the first drive. For partition 2 on the first drive use **divvy /dev/hd02**. Refer to the **hd(HW)** manual page for a table of hard disk device nodes.

Because the **divvy** filesystem table is not located in the super block, the information in the table is probably correct. From the **divvy** table, obtain the first and last block number of the filesystem that you are attempting to restore.

2. To obtain the correct value for **FSIZE**, apply the following formula:

**FSIZE = last\_block - first\_block + 1**



3. To obtain the correct value for **ISIZE**, run **bc(C)**, and enter the following command:

$((\text{FSIZE} * .25) - ((\text{FSIZE} * .25) \% 16))$

where **FSIZE** is the value computed in step 2. Round down any fractions and record this calculation of **ISIZE** for later use. Exit **bc** by entering **quit**.

4. To convert the number of available inodes into the actual address of the first block following the blocks allocated for inodes, you must perform an additional calculation. Refer to this number as **isize** to distinguish it from the **ISIZE** value displayed when **fsdb** starts.

To do this, perform the following calculation:

$\text{isize} = (\text{ISIZE} / 16) + 2$

where **ISIZE** is the number that you calculated in step 3; you should round the value as previously described.

5. With your **FSIZE**, **ISIZE**, and **isize** values, you can now correct the super block values with **fsdb(ADM)**. Enter the following:

**fsdb /dev/filesystem**

where *filesystem* is the name of the filesystem that you want to repair.

6. If the values that you calculated are close approximations of the real numbers, **fsdb** displays output similar to the following:

**FSIZE = 52985, ISIZE = 13232**

If the values match your calculation, you do not need to modify that parameter. If both values seem correct, something else is wrong and you must restore your data from backups.

**NOTE** You can end the display of addresses at any time by pressing **INTERRUPT** or **(Del)**.

7. Decide which parameter (or both), **isize** or **FSIZE**, must be corrected.
8. Enter the following commands (for XENIX, or UNIX system and AFS filesystems), substituting the values that you calculated earlier for **FSIZE** and **isize**. In the instructions that follow, your input is in bold and **fsdb** responses are in normal font, with real values in place of xxx and yyy. Remember to use the actual values for **FSIZE** and **isize**; do not enter the words "FSIZE" and "isize".

If **isize** is correct, simply skip it by pressing **(Return)**.

**NOTE** Each time you press **(Return)**, **fsdb** displays what the filesystem currently thinks the values of **FSIZE** and **isize** are.

To repair a XENIX filesystem, enter these commands:

```
1024(Return)
002000: 000000 (0)
=i size(Return)
002000: 000xxx
(Return)
002002: 000000 (0)
=F SIZE(Return)
002002: 000yyy
q
```

To repair an S51K, EAFS or AFS filesystem:

```
512(Return)
001000: 000000 (0)
=i size(Return)
001000: 000xxx
(Return)
001002: 000000 (0)
(Return)
001004: 000000 (0)
=F SIZE(Return)
001004: 000yyy
q
```

9. If the corrupted FSIZE value is so small that **fsdb** thinks that you cannot move that far into the block, the following error message is displayed when you input the beginning address:

block out of range

To disable error checking, enter a capital "O". You can then input the address without complaint.

At this point, you can run **fsck** on the filesystem and properly restore the system.

**NOTE** Currently, you cannot use **fsdb** to repair a DOS filesystem.

### *Filesystem mount failed*

If the **mount(ADM)** command fails, check the filesystem with the **fsck(ADM)** command before running **mount** again:

```
/etc/fsck -y /dev/filesystem
```

where *filesystem* is the name of the filesystem that you want to check and repair.

## Unable to remove files

---

If a user has write permission on a directory but is unable to remove files from that directory, the *sticky bit* has been set for that directory. The sticky bit is a directory protection setting that allows only the owner of the file (or *root*) to remove files from that directory.

**NOTE** The contents of the files in the directory can be modified or removed, if the permissions allow write access.

Only *root* or the owner can set the sticky bit; only *root* or the owner of the directory can remove it.

To determine whether a directory has the sticky bit set, do a long listing of its parent directory (the sticky bit shows up as a “t” in the last field of the permissions listing). The following example shows a directory on which the sticky bit has been set:

```
drwxrwxrwt 5 sys sys 2432 Jan 17 16:58 tmp
```

To remove the sticky bit setting from a directory, enter the following command:

```
chmod -t directory_name
```

Users should now be able to remove files from the directory.

## Login problems

---

This section covers what to do in situations in which:

- you cannot log in to multiscreens on the console after the system enters multiuser mode
- users cannot log into the system
- the system displays “Login incorrect” when a user tries to log in
- a user’s account is locked
- a user forgot their password

### Cannot log in after entering multiuser mode

---

If, after entering multiuser mode, you cannot log into the console even as *root* and the system displays the following error message:

```
Cannot obtain database information on this terminal
```

Refer to “Cannot obtain database information on this terminal” (page 978) for more information.

## ***Users cannot log into the system***

---

When the `/etc/group` file is missing, users cannot log in and the system displays the following message on the user's terminal:

```
Can't rewrite terminal control entry for tty01.
Authentication error; See Account Administrator
```

The following error message is also displayed at boot time:

```
/bin/su:cannot setgid to auth, no auth entry
```

Refer to "Can't rewrite terminal control entry for tty" (page 978) for more information.

## ***Login incorrect***

---

If the system displays the "Login incorrect" error message when a user tries to log in, one of several problems may exist. Refer to "Login incorrect" (page 978) for more information.

## ***useshell: file access control database inconsistency***

---

If the system displays messages regarding "useshell" when creating or modifying a user, permissions on certain system files are incorrect. Refer to "useshell: File Control database inconsistency" (page 979) for more information.

## ***Unlocking a locked user account***

---

If a user account has been locked deliberately by the system administrator, the system locked the account because a user exceeded the number of unsuccessful logins attempts, or the user's password is expired, the system displays the following message when a user tries to log in to that account:

```
Account is disabled -- see Account Administrator
```

To unlock the account, invoke `sysadmsh` and select:

```
Accounts ⇨ User ⇨ Examine:Logins
```

Change the "Lock status" field to "Clear all locks". For more information, see "Locking or unlocking a user account" in the chapter on administering user accounts in *Administering Operating System Services*.

## ***Replacing a forgotten user password***

---

If a user forgets their password, the system administrator must change the password to a new one. The system does not provide a way to decipher an existing password.

To do this, follow these steps:

1. Invoke **sysadmsh** and make the following selection:  
Accounts ⇨ User ⇨ Examine:Password
2. Select “Change” from the Current password status selections.
3. Choose “Yes” from the Confirm Change form to select a new password. This invokes the password change procedure.
4. Select “1” (to select a password) or “2” (to have the system generate a password) and enter or choose a new password.
5. Press ⟨Return⟩ to return to the Modify User Account form.

For more information, see the chapter on administering user accounts in *Administering Operating System Services*.

## Mail troubleshooting

---

The MMDF system is configured for the local system by default. Therefore users should not experience problems with local mail. Generally, when mail does not work, there is either a problem with the network, or with MMDF configuration.

When mail problems occur, first verify that the network is working. To do this, test the appropriate network (for example, UUCP or TCP/IP). For information on testing the UUCP connection, see the chapter on building a remote network with UUCP in *Administering Networking Services* and “UUCP troubleshooting” (page 985). See your networking documentation for other testing information.

If UUCP is working and you still experience problems with mail, the problem is most likely with the MMDF configuration or the deliver daemon. The following sections discuss some common problems and solutions.

### Failed mail error

---

If the system returns your mail immediately in a mail message with the words “Failed Mail” in the Subject line, there is a problem with the channel (\*.chn) or domain (\*.dom) files in the */usr/mmdf/table* directory. Use the following procedure to check this:

1. Log in as the **mmdf** user.
2. Change directories to */usr/mmdf/bin*.

3. Invoke **checkaddr** with the mail address that you used. For example, if **mail sylvia@seattle** failed, enter:

```
/checkaddr sylvia@seattle
```

4. If **checkaddr** displays "Unknown Host Domain," either the host or domain is not in the channel or domain files, or the */usr/mmdf/mmdftailor* file contains an error.

If **checkaddr** displays "OK," then the host and domain are in the channel and domain files, but are incorrectly configured. Check the address with **checkaddr -w** to diagnose this problem.

5. Using the information in the chapter on setting up electronic mail in *Administering Networking Services*, verify that your domain, channel, and */usr/mmdf/mmdftailor* files are set up correctly.

## ***Mail does not work, no returned mail***

---

If mail does not reach its destination but MMDf does not return unsent mail, either **deliver** is not running properly, or the */usr/mmdf/mmdftailor* file contains an error. The **deliver** daemon is started automatically by a script in the */etc/rc2.d* directory at system startup; the process is owned by **mmdf**. Check to see that **deliver** is running using **ps -ef**. If **deliver** is not running, use the following procedure:

1. Log in as the **mmdf** user.
2. Start the **deliver** daemon manually with the following commands:

```
cd /usr/mmdf/bin
```

```
/deliver [-channelname] -b -T60
```

where *channelname* is the channel that you want to check (for example, UUCP or SMTP); if you do not specify a *channelname*, **deliver** checks all channels. This starts **deliver** for the specified channel sweeping the mail queue in the background every 60 seconds. For more information, see the **deliver(ADM)** manual page.

3. If running **deliver** manually does not work, check the configuration of the domain and channel files in */usr/mmdf/table* and the */usr/mmdf/mmdftailor* file.

## ***Mail command hangs***

---

If the **mail** command hangs when a user tries to read mail, check the following:

1. Verify that the user's mailbox exists in the */usr/spool/mail* directory. Unless your postmaster has configured the system otherwise, the */usr/spool/mail* directory should contain a file for each user who receives mail.

2. Make sure that the owner and group ID of the `/usr/spool/mail` directory is `mmdf`.
3. Make sure that the mailbox is not too large (1000 messages is regarded as too large).

## Modem problems

---

This section discusses common problems that you may have with your modem. Note that, while other serial ports are often used, the examples in the modem troubleshooting sections assume that the modem is attached directly to COM1. If you have problems, first verify that the phone jack is plugged in and that you have a dial tone on the phone line.

## Modem — errors when dialing out

---

This section describes some situations and solutions to problems that may occur when dialing out on your modem.

The most useful tool for diagnosing dial-out problems is the `-x9` option to `cu(C)`. This option causes `cu` to display diagnostic output when attempting to dial out. To get a debugging output, enter the command:

```
cu -x9 phone_number
```

where *phone\_number* is the phone number of the system you wish to dial.

### No OK message (Hayes-compatible modems)

---

You get a connected message when you test the modem connection with the command:

```
cu -s1200 -l tty1a dir
```

but, when you enter `AT`, the system does not display the “OK” message. Use the following steps to solve the problem:

1. Verify that the modem switch and software settings are correct.
2. Check the modem cable:
  - If you are using a straight-through cable, try a null modem cable using at least pins 2, 3, 7, 8, and 20.
  - After issuing the `cu` command, watch the lights on the modem and press `(Return)` several times. The “receive” light should flash as you hit `(Return)`. If it does not flash, check your cable to make sure that pin 2 is connected correctly (pin 2 is the data transmission line from the serial port to the modem).

3. If the “send” light flashes on the modem, the local echo may be turned off. Use the ATE1 command to turn on the modem’s echo capability.
4. Verify that the serial port on the computer is not defective:
  - Attach the modem to a different serial port, or attach a terminal or serial printer (with a proper cable) to the port to confirm that it is functioning.
  - If the port is not functioning, check your hardware documentation for an appropriate repair facility.
5. If the previous steps do not fix the problem, your modem may be defective. If this is the case, check your hardware documentation for an appropriate repair facility.

### ***Modem dials, but does not connect***

---

If your modem dials okay, but the call never connects, check the following:

1. Verify that the phone number is correct and operational and that the phone line to which the modem is attached is not faulty. To do this, unplug the modem from the telephone line and plug in a regular telephone. Manually dial the number to make sure that the modem on the other end of the line is answering the call.
2. Listen carefully to your modem while it dials the call. Some business phone systems require a pause between certain numbers. Use a hyphen in the cu command to indicate a pause of two seconds. For example, if you enter: 9----458--1234, the modem pauses 8 seconds after dialing the first number.

The dialer translates the hyphen passed to the cu command into the appropriate code for your modem. For example, the dialer translates the hyphen into a comma before sending to a Hayes-compatible modem.

### ***Connect failed: NO DEVICES AVAILABLE***

---

When you try to dial out on the modem, the following message is displayed:

```
Connect failed: NO DEVICES AVAILABLE
```

Follow these steps to solve the problem:

1. Verify that the modem port has an entry in the */usr/lib/uucp/Devices* file. Here are example entries for a Hayes-compatible modem running at 2400 baud on */dev/tty1A*:

```
ACU tty1A - 300-2400 /usr/lib/uucp/dialHA24
Direct tty1A - 2400 direct
```

Make sure that these lines in *Devices* do not begin with a number sign (#). There should be no spaces in front of the entries, and the “ACU” line must come before the “direct” line or getty will not hang up the modem correctly.



2. Verify that the modem port in *Devices* has the correct baud rate associated with it. If you specify the baud rate with the `-s` option to `cu`, verify that there is an entry in *Devices* that corresponds to that baud rate.

## *Modem answers, but terminal displays garbage*

---

If the modem answers, but the terminal displays garbage characters, use the following steps:

1. Verify that the site that you are calling is set to the same data bit and parity values that you are using. By default, `cu` uses 8 data bits, and no parity. To change the values to 7 data bits and even parity, enter `cu -e`. For 7 data bits and odd parity, use `cu -o`, and use `cu -oe` for 7 data bits and no parity.
2. Verify that the remote computer is set to the same baud rate that you are using.

If you are dialing into another UNIX system, you can force the remote site to switch to the next baud rate by sending a break signal. To send the break signal during the login sequence, enter:

`~%b`

3. Check for noise on your phone line. Noise problems become more acute when operating at 2400 baud or higher. Normally, when there is a problem with line noise, garbage characters appear on the screen in short bursts or continuously, as if a system on the other end of the line is trying to send valid data.

## *DEVICE LOCKED*

---

Refer to “DEVICE LOCKED” (page 992) for more information.

## *Modem does not hang up*

---

If your modem does not hang up at the end of a call, check the following:

1. Verify that you are using a modem control port that is configured in the `/usr/lib/uucp/Devices` file and that the “ACU” entry for the modem port comes before the “Direct” entry for the direct line. If you are using a non-modem control port, change the port to the corresponding modem control port. For example, the modem control port associated with `tty1a` is `tty1A`.

**NOTE** Non-modem control ports should only be used with terminals, and when configuring the modem.

2. If the CD (Carrier Detect) light on the modem does not go off when the call is disconnected, check the modem switches to verify that the modem is set to detect the incoming carrier. If your modem is a Hayes 2400 or

compatible, use the `AT&C1` command. This forces the carrier detect line to follow the presence of a carrier on the phone line.

3. Check the modem switches to verify that the modem is set to detect DTR (Data Terminal Ready). The modem should hang up when DTR goes from high to low. If the modem is a Hayes 2400 or compatible, use the `AT&D2` command.
4. Some modems have a switch that can be set to ignore DTR; make sure that this switch is off.

### *Double echo*

---

If you get a double echo when you dial out on your modem, check the setting for local echo. If local echo is enabled, disable it.

### *Modem — problems dialing in*

---

This section provides solutions for common problems that may occur when dialing in on your modem.

#### *Modem does not answer the phone*

---

If the modem does not answer the phone, check the following:

1. Verify that the modem control port is enabled. To enable the modem port, enter the following commands:

```
disable /dev/tty1a
enable /dev/tty1A
```

2. Verify that the modem is configured to auto-answer. Check your modem switches. If the modem is a Hayes 2400 modem, enter:

```
cu -l tty1a dir
```

Then, use the `ATS0=1` command to tell the modem to answer the phone on the first ring. (Remember to enter `AT&W` to save modem settings.)

3. Verify that the DTR (Data Terminal Ready) line is connected from the computer to the modem. Make sure that Pin 20 is connected. Pins 2, 3, 7, 8, and 20 are required for modem communication.
4. Make certain the ACU entry for this modem in the *Devices* file precedes any "Direct" entries for the port or the non-modem control counterpart.

## Modem answers, but hangs up

---

If the modem answers, but hangs up immediately upon connection, check the following:

1. If the modem is set to auto-answer and to detect DTR, check to see that the DTR line is asserted.
2. Verify that the modem control port is enabled:  

```
disable /dev/tty1a
enable /dev/tty1A
```
3. Verify that the cable is correct. If you are using a straight-through cable with at least pins 2, 3, 7, 8 and 20 connected, verify that pin 20 (DTR) is properly connected.
4. Check to see if the modems have data compression or error correction modes set. After a connection is established, modems that support special error correction or compression protocols attempt to “negotiate” which of these protocols to use. If the other modem is programmed not to use any of these modes (or does not support them), it may misinterpret this negotiation as actual user input. In general, modes such as MNP4 or v.42bis should only be enabled when connecting with a modem with the same modes enabled. Check the documentation for your modem.

## Garbage or loose cable

---

The console displays a message like the following when a call comes into the modem:

```
Garbage or loose cable on /dev/tty1A, port shut down
```

Check the following:

1. Verify that your modem is not set to echo back data or send command responses. If the modem is not set up this way, it may be sending a “RING” signal to indicate that the phone you are calling is ringing. Because the CD signal is not active, the serial driver interprets this as random data on the serial line. The proper Hayes 2400 modem command is ATE0Q1.
2. If you have an internal modem and the above options do not eliminate the error message, your modem may be incompatible. Contact the manufacturer to see if a fix is available. If no fix is available, you may need to replace your modem with a standard Hayes-compatible external modem.

## ***Modem answers, but no login prompt***

---

If the modem answers, but does not display a login prompt, check the following:

1. Verify that the CD line is being asserted by the modem after the modem has answered the phone. Check the switches on your modem or, if your modem is a Hayes 2400 or compatible, use the AT&C1 command. (Remember to enter AT&W to save modem settings.)
2. Make sure that the port is enabled. Enable the port by entering the following command sequence:

```
disable /dev/tty1A
enable /dev/tty1A
```

3. Verify that the modem is using the correct */etc/gettydefs* entry and is selecting the proper baud rate. The modem port device line in the */etc/inittab* file should look like the following:

```
t1A:2:respawn:/etc/getty -t60 tty1A 3
```

The last field on the line is the pointer to the entry in the */etc/gettydefs* file. In this case the */etc/gettydefs* entry is "3", which is the 2400-1200-300 baud entry. Verify that this entry in */etc/gettydefs* is correct.

## ***Screen displays a series of login prompts***

---

If the screen scrolls uncontrollably when you log in, usually displaying a series of login prompts, verify that only the modem device is enabled. If the non-modem device is enabled, disable it:

```
disable /dev/tty1a
```

## ***System displays meaningless characters***

---

If the system displays the login prompt, but no password prompt, or meaningless characters are displayed after the login prompt, verify that the line settings are correct:

1. Determine the serial line settings on the system that you are calling. The standard settings that cu uses are 8 data bits, one stop bit, and no parity.
  - If the remote system uses even parity, use cu -e.
  - If the remote system uses odd parity, use cu -o.
  - If the remote system uses 7-bit no parity, use cu -oe.
2. If you are dialing into a UNIX system, check the */etc/inittab* file on the remote system to verify that the "pointer" into the */etc/gettydefs* file is correct. The serial line characteristics may not match between the stty

settings defined in the third field of the selected *gettydefs* entry. Change the setup for the port to 8 data bits, one stop bit, and no parity.

The entry should be similar to the following:

```
3 # B2400 HUPCL OPOST CR1 ECHOE NL1 #
 B2400 CS8 SANE HUPCL TAB3 ECHOE IXANY #\r\n@!login: # 1
```

## Mouse (serial) troubleshooting

---

The error messages in this section are associated with mouse problems. If you only recently installed the mouse, make certain you checked all the points covered in the chapter on adding mice and other graphic input devices in the *Hardware Configuration Guide*.

### *usemouse: no mouse available(5)*

---

Try executing the `tty(C)` command on the multiscreeen that is failing. If the `tty` command is executed on the first multiscreeen, it will normally return `/dev/tty01`. If `tty` returns `/dev/syscon` on the failing multiscreeen then the system was shutdown on that multiscreeen using the `init s` or `init S` command. Shutting down the system with `init s` or `init S` will cause the system to enter single-user mode. As documented in the `init(M)` manual page, this will also cause the system console device (`/dev/syscon`) to remain linked to the `tty` from which the last `init S` was invoked.

To correct this problem, and allow the mouse to work, take the following steps:

1. Log in as *root* and run `mkdev mouse`.
2. Select option 4, "Associate a terminal with an existing mouse."
3. Enter `syscon` as the terminal device.
4. Enter `exit` to exit from `mkdev mouse`.

This adds `/dev/syscon` to the list of terminals with which the mouse can be used.

### *usemouse: can't open slave (errno 13)*

---

This message means that `usemouse` cannot open the slave side of the pseudo-`tty` that is necessary to communicate with the system.

Check the permissions on the "master" pseudo-`ttys` (devices beginning with `/dev/ptyp`) and "slave" pseudo-`ttys` (devices beginning with `/dev/ttyp`). They should be 666 mode or "rw-rw-rw-". If they are not correct, change them using the `chmod(C)` command. As a general rule, if a problem is related to permissions, running as a normal user gives the problem, but running as *root* does

not. In addition, check the entries in the file `/usr/lib/event/ttys` which indicate that a mouse is a legitimate “event source” for the tty you are on. For example:

```
tty01 msbusmouse
```

This entry indicates a Microsoft busmouse is configured as usable on the first console multiscreen.

### *Mouse (X server) troubleshooting*

---

The X server goes into graphics mode before the user can see all the error messages it might display. This is particularly evident if you do not have the mouse correctly installed or configured. If the server processes (such as `scolgin(Xm)` and `startx(Xm)`) do not start, issue the following command to execute the X server:

```
$ /usr/bin/X11/X 2> /tmp/xsrrvr.errors
```

If the server fails, the error message is recorded in the `xsrrvr.errors` file. Look at this file; the following message:

```
Open event driver failed
```

indicates that there was a problem with the mouse. Run `mkdev mouse` to remove and reconfigure your mouse; this usually resolves the problem.

### *No more processes error message*

---

The “No more processes:” error message is displayed when you attempt to execute a process that would exceed some system resource. This error message can indicate one of several conditions, including:

- The memory required to run all the user processes is more than the total available main memory and swap space. At least 10% of available memory should be free at any given time. To check this, use the `crash(ADM)` command. Within `crash`, get the values of the `availsmem`, `nswap`, and `maxmem` kernel variables by using the `od -d variable` command. The following equation determines the percentage of usable memory that is still available:

$$\text{availsmem} / ((\text{nswap} / 8) + \text{maxmem})$$

This quotient should be greater than 0.10 at all times. Larger numbers are better, especially on systems that are running X and Motif clients. See “Understanding your performance goals” (page 869) for a discussion of ways to resolve this problem.

- You have run out of entries in the process table whose size is determined by the `NPROC` tunable parameter. The `sar -v` command gives statistics for the process table.

- You are running more processes than the system allows per user. Try scheduling some jobs to run at a different time. If this is not possible, you may need to adjust the value of the **MAXUP** tunable parameter. See **MAXUP** (page 916) for more information.

## Runaway processes

---

A runaway process is a process that enters an infinite loop and spawns new processes. This can cause an overflow in the **proc** table that causes other processes to fail with the "No more processes:" error message. If the **sar -v** report shows that the kernel **proc** table has overflowed, use the **ps(C)** command to check for a runaway process. To display all the processes on the system, use the **ps(C)** command, such as the following:

```
ps -ef | grep -v getty | more
```

This displays all executing processes except for **getty** lines that represent idle terminals. Look for processes that have consumed large amounts of time, or several similar processes that may have been spawned by a runaway process that is in an infinite loop.

See "Runaway processes, stopping" (page 974) for information about how to stop a runaway process.

## Print system troubleshooting

---

This section covers problems you might encounter while using your printer.

### *lpsched print scheduler is not running*

---

If the print service stops in the middle of a print request, or does not start any new print jobs, determine if the **/usr/lib/lpsched** daemon is running. To do this, enter the following command:

```
lpstat -r
```

(The **lpsched** daemon is started automatically by a script in the **/etc/rc2.d** directory at system startup.)

If the scheduler is down, the system displays the following message:

```
scheduler is not running
```

If this is the case, use these steps to start the **lpsched** daemon:

1. Log in as the super user **root** or a user with **lp** authorization.
2. Invoke **sysadmsh** and select **Printers ⇄ Schedule ⇄ Begin**. The system displays the following message:

```
Print services started
```

It may take a minute or two for the printer configurations, forms, and filters to be re-established before any saved print requests start printing. Any print

requests that did not finish printing when the scheduler stopped are printed in their entirety when the print service restarts. The printer configurations, forms, and filters in effect when the print service stops are restored when it restarts.

### *Printer does not print*

---

If the printer is sitting idle and there is no output, check the following:

1. Make sure that the printer has power.
2. Check that the printer is not out of paper, ink, or toner and that it is not in an error state.
3. Verify that the printer hardware is working before continuing. Check the printer documentation for a self-test.
4. Check the printer cable and make sure that it is attached properly to the port and the printer. Refer to the owner's manual for your printer for installation instructions.
5. Make sure that the printer is configured properly. To set up your parallel or serial printer to receive data properly, follow the instructions for installing a printer in the *Hardware Configuration Guide*.

If the printer is a serial printer, make sure that the baud rate at which the computer sends data to the printer matches the printer's baud rate. For instructions on how to reset the baud rate, see "Printer output is illegible" (page 971).

6. Make sure that the printer is enabled. To do this, invoke `sysadmsh` and select:

Printers ⇌ Schedule ⇌ Accept

Printers ⇌ Schedule ⇌ Enable

7. Verify that the system has the port configured properly. Use the `hwconfig(C)` command or check the `/usr/adm/messages` file for these messages (for parallel and serial, respectively):

```
%parallel 0x378-0x37A 07 - unit=0
```

```
%serial 0x03F8-0x03FF 04 - unit=0 type=Standard nports=1
```

If you do not see a similar message for the printer port, refer to "Port does not respond" (page 970) for more information.

8. Make sure that the port is configured for the proper interrupt vector and that no other hardware is using that interrupt vector. Refer to the *Hardware Configuration Guide* for information on the available interrupt vectors. See your hardware documentation for information on configuring your ports.



9. Test the printer port connection by redirecting the output of a command directly to the device.

- For parallel printer *lp0*, enter:

```
date > /dev/lp0
```

- For serial printer *tty1a*, enter:

```
(stty options;date) > /dev/tty1a < /dev/tty1a
```

where *options* are baud rate, parity, or other settings that you want to pass to the serial printer. These options should normally include *opost* and *onlcr*.

If the output from the redirected command does not print, follow the instructions in “Cannot redirect output to printer” (this page).

If the output prints, try submitting a sample file (like */etc/motd*) for printing by invoking *sysadmsh*, and selecting:

Dirs/Files ⇌ Print

If the hardware connections are good and the printer is properly configured and enabled, but is still idle and print requests are queued, check the following:

1. Issue the *lpstat -r* command to verify that the *lpsched* process is running.
2. Restart the *lpsched* daemon if it is not present:

```
/usr/lib/lpshut
/usr/lib/lpsched
```

3. Check to see that print requests are being queued:

```
lpstat -o -l
```

This command displays a detailed description of the status of output requests, printer names, and devices.

4. If the printer detects a fault, it does not immediately continue automatic printing. Force a retry by enabling the printer.
5. Check to see if a dialout printer was busy or did not answer, or all dialout ports are busy. The print service waits five minutes before trying to reach a dialout printer again. Force a retry by enabling the printer.

## Cannot redirect output to printer

If you redirect output directly to the parallel or serial port and nothing happens or the system displays the “cannot create” message, check the following:

1. Verify that the device file for the port exists in */dev*. Make sure that this file is a device file and not a text file. For example, use the following command to check *lp0*:

```
ls -l /dev/lp0
```

The output should look similar to this:

```
crw----- 2 bin bin 6, 0 Jun 13 1992 /dev/lp0
```

A text file would be missing the "c" at the beginning of the line.

2. Test the cable connection using a cable from a working with the same cabling requirements.
3. Print a file while the system is running under DOS. If you can print a file under DOS but not under the UNIX system, check the following:
  - Verify that the port is configured correctly; see "Port does not respond" (this page) for more information.
  - If the port configuration is correct, and you still cannot redirect output to the port, try using a different device name. For example, for a parallel port, use *lp1* instead of *lp0*; for a serial port, use *tty2a* instead of *tty1a*. Pay attention to the "unit=" message for parallel ports on the boot screen, or by using *hwconfig* (for example, *unit=0* corresponds to *lp0*).
  - If you still cannot print using a different device name, your printer may be defective; check the hardware documentation that came with your printer.
4. If you cannot print from DOS, check the printer hardware configuration. See the documentation that came with your printer.

If you configured your printer correctly and you still cannot redirect output to it, the problem is most likely a hardware malfunction. Recheck the cables and port configuration and consult your hardware documentation.

### *Port does not respond*

---

If your serial or parallel port does not respond, check the following:

1. Verify that the parallel or serial card is properly seated:
  - Turn the power off and open the machine.
  - Remove and reseat the card in the bus.
2. Run *mkdev serial* or *mkdev parallel* to see if the port is listed in your kernel configuration.
3. Verify that the card itself is correctly configured; check the documentation that came with the card. If possible, try setting the card for a different configuration.
4. The card may be defective; replace it.

## Printer output is illegible

If the printer prints illegible output, check the following:

1. Determine the baud rate for the serial printer and check to see that it matches the baud rate for the computer. (No baud rate is associated with a parallel port, although `stty(C)` does display one.) Set the baud rate to 9600 baud for optimum performance.

To set a different baud rate for the print service to use, use the following command:

Printers ⇨ Configure ⇨ Parameters

In the “Default initial settings” section, enter the baud rate number in the “`stty`” field. Then submit a sample file for printing.

2. Determine the parity setting for the printer and check to see that it matches the computer’s parity setting. (If the printer is directly connected to the computer with a wire that is less than 50 feet long, it does not have to use the parity bit.)

To set the parity bit, use the following `sysadmsh` selection:

Printers ⇨ Configure ⇨ Parameters

In the “Default initial settings” section of the form, add one of the following to the “`stty`” field:

`oddp` Sets odd parity generation.  
`evenp` Sets even parity generation.  
`-parity` Sets no parity (default).

Select the option that matches what your printer requires.

3. Check to see that the tabs are set correctly. Refer to “Printer output spacing is wrong” (page 972) for more information.

If the settings and baud rate are correct and the output is still illegible, check to see that the printer type is correct. If the wrong printer type was selected when you set up the printer with the print service, the wrong control characters can be sent to the printer. This situation can cause output to disappear or be illegible. A simpler problem to solve is when it sets the wrong character set or font.

If you do not know the printer type, examine the `/usr/lib/terminfo` directory for a list of names that are available. Look for names that match one word or number identifying your printer. For example, the name 495 identifies the AT&T 495 Printer.

When you have the name of a printer type that you think is correct, change the printer type setting. To do this:

1. Invoke `sysadmsh` and select:  
Printers ⇌ Configure ⇌ Parameters
2. In the printer name (terminfo database) field, enter the printer type name.

### ***Printer output spacing is wrong***

---

If the printer output is legible, but the spacing is wrong, invoke `sysadmsh` and select:

Printers ⇌ Configure ⇌ Parameters

For each of the situations, adjust the following settings in the “`stty`” field of the “Default initial settings” section:

- If the printer output is double-spaced, enter either the `-onlcr` or `-tabs` option.
- If there is no left margin and the text runs together, enter the `-tabs` option.
- If the printer output zig-zags down the page, enter the `onlcr` option. (This is set by default, but you may have cleared it accidentally.)

### ***Parallel printer is slow***

---

If your parallel printer prints abnormally slowly, verify that the configuration settings are correct using the the instructions for installing a printer in the *Hardware Configuration Guide*. In particular, printing can be slow if the port is configured for the right I/O address but the wrong interrupt vector. To check this, run `mkdev parallel` and note the interrupt used by the slow port. Use the `remove` option, and then add it once again with the same I/O address, but switch the interrupt (5 instead of 7, or vice versa). Relink, reboot, and see if the port is still slow.

If printing is still slow after verifying that the parallel ports are configured correctly, the problem may be that your parallel port is not capable of generating interrupts. In this case, you can try setting up polling for your printer port. If your printer is taking about four seconds per line, the printer may be deselecting itself after receiving each line of text and then reselecting. In this case, there is a kernel parameter that changes this behavior. Each solution is described in the sections that follow.

### ***Setting up polling***

To speed up printing on your parallel printer, you can alter the way that the hardware and the printer driver communicate. The parallel printer driver can be made to “poll” a parallel port so that the driver does not rely on interrupts from the parallel port.

**NOTE** When the printer driver polls a parallel port, you may experience a drain on system resources.

To set up polling for a parallel port or parallel printer, create what is known as a *special device node*.

Use the following procedure:

1. Use the **shutdown su** command to bring the system into maintenance mode.
2. Note which parallel printer ports are configured. (Check `/usr/adm/messages` or run the `/etc/hwconfig -h` command.) For example:

```
parallel 0x378-0x37A 07 ~ unit=0
```

where "unit=0" refers to `lp0`.

3. Configure a special device file for the printer by creating a file called `pa` (if it does not already exist) in the `/etc/conf/node.d` directory containing one of the following lines:

```
For lp0: pa lp0p c 64 bin bin 600
For lp1: pa lp1p c 65 bin bin 600
For lp2: pa lp2p c 66 bin bin 600
```

4. Enter the following commands:

```
cd /etc/conf/cf.d
touch /etc.new_unix
../bin/idmkenv
```

5. You see the following:

```
The kernel environment includes device node files and /etc/inittab.
The new kernel may require changes to /etc/inittab or device nodes.
Do you want the kernel environment rebuilt? (y/n)
```

Enter "y" and press (Return).

6. The following is displayed:

```
The kernel has been successfully linked and installed.
To activate it, reboot your system.

Setting up kernel environment
```

7. Enter the **haltsys** command to reboot your system. Reboot the system as instructed and enter multiuser mode. The new polling device is now in place.

8. If you are using the print spooler, you must now inform the spooler of the new parallel poll device. Invoke `sysadmsh` and select:

Printers ⇨ Configure ⇨ Modify

9. In the “Device name” field, do not select a standard parallel device name. Instead, use either `/dev/lp0p`, `/dev/lp1p`, or `/dev/lp2p`.

### *Changing the MODE\_SELECT parameter*

If you suspect that your printer is deselecting and reselecting itself after each line, you can change the value of the `MODE_SELECT` kernel parameter. (This works with both polled and standard `lp` devices.) `MODE_SELECT` has a default value of 1 which enables mode-select checking. Setting this value to 0 disables mode-select checking, and hence removes the printing delay incurred by these checks.

To change the value of `MODE_SELECT`, use the `configure(ADM)` utility. The `MODE_SELECT` parameter is found in the “Hardware dependent parameters” category of the `configure` menu. See “`configure(ADM)` — reallocate kernel resources” (page 882) for complete instructions.

**NOTE** Slow printing has been observed on some models of Tandy® printers, but is not found exclusively on Tandy printers.

## *Printer reports UUCP errors*

---

If UUCP is configured, the print service uses the UUCP software to handle dialout printers. If a dialing failure occurs and you receive printer fault alerts, the print service reports the same error reported by the UUCP software for similar problems. (If you have not arranged to receive fault alerts, this information is mailed to the user `lp`, by default.) Refer to the section on UUCP error messages in the chapter on building a remote network with UUCP in *Administering Networking Services* and “UUCP troubleshooting” (page 985) for more information.

## *Runaway processes, stopping*

---

A runaway process can cause an error that locks up the keyboard, preventing anything that the user types from reaching the system. Because of this, a runaway process cannot be stopped from the terminal at which it was invoked.

To continue, you must identify and stop the runaway process:

1. Log in as `root` on a terminal (or console screen) that is not locked.

2. Enter the **ps -ef** command and press **<Return>**. The system displays full information on all the current processes and their process identification numbers (PIDs). Find the PID of the runaway program. You may see on program that is consuming more CPU time than is reasonable, or may find a large number of processes with a common parent who are running the same program.
3. Enter the **kill PID** command. The program should stop in a few seconds. If the process does not stop, enter the **kill -9 PID** command.

The last command may leave temporary files that are usually removed when a program terminates normally, or a non-echoing terminal. Refer to “Restoring a non-echoing terminal” (page 985) to restore the terminal to normal operation. If the runaway process does not stop when you enter **kill** with the **-9** option, the process is considered *unkillable*. If the unkillable process is a user’s shell, you must stop the process before that user can continue working. To stop an unkillable process, you must reboot the system. Use the following procedure:

1. Log in as *root* and send a message using **wall(ADM)** to the other system users notifying them of the impending shutdown.
2. When all the users have logged out, shut down the system by entering **init 6**. The root prompt (**#**) is returned, but the shutdown process begins soon afterward.
3. Reboot the system by pressing **<Return>** at the **Boot:** prompt.

The unkillable process no longer appear when you run **ps**.

## Security system troubleshooting

---

This section lists problems and error messages that you may encounter. Each problem is discussed in context, including the reason for the situation, the solution to the problem, and ways to prevent the situation from recurring. See the chapter on administering user accounts in *Administering Operating System Services* in *Administering Operating System Services* for more information about administering the security software.

You must log in as *root* to fix the problems discussed here. In some cases, you must use the “override” terminal to fix problems with the security system. This feature is discussed first.

### Using the override terminal

---

If your system is running with High security defaults, an override terminal entry exists for *root*. This terminal can be used if the security databases become corrupted so that all logins are disallowed. The */etc/default/login* file

contains an entry that identifies which terminal (tty) is to be used for an override login for *root*. The default entry (shown below) permits *root* to log in on */dev/tty01*, also known as the first multiscreen on the console.

```
OVERRIDE=tty01
```

You can change this default to be another login device.

When the databases are compromised and *root* logs in on the override terminal, the following message is generated:

```
The security databases are corrupt.
```

```
However, login at terminal tty is allowed.
```

When the account is locked and *root* logs in on the override terminal, the following message is generated:

```
Account is disabled but console login is allowed.
```

The device used as the override terminal should be physically secure; remember that normal locks do not apply to the super user account on this tty.

If you are locked out and you removed the default override tty, you must reset the system or power cycle it, and come up in single-user (maintenance) mode to enter the system. Because shutting off the system can lead to file-system damage, it is critical that you have an override terminal.

### ***Account is disabled – see Account Administrator***

---

If you see this message, it means the account is locked for one of three reasons:

- You locked the account through the `sysadmsh` selection Accounts ⇨ User ⇨ Examine:Logins. If you want to re-establish the account, use the same selection to unlock the account.
- The password lifetime for the account elapsed. The password for the account did not change before the password lifetime was over. To re-enable the account, you can either assign a new password or use the Accounts ⇨ User ⇨ Examine:Logins selection to clear the lock (in which case the user is forced to set the password at login time). Advise users to change their passwords before the lifetime expires.
- A number of unsuccessful tries were made on the account, exceeding the threshold number you set for locking it. These tries may not have all been made on the same terminal. Before re-enabling the account, it is a good idea to determine the cause for the lock-out. It may be that the user is a poor typist: or another person is trying to lock the account and knows this is a way to do it, or a real attempt to penetrate the account is being made. You may want to adjust the threshold upward or downward, depending on the nature of the system users, the value of the data, and the accessibility of the system to outsiders.



## ***Account is disabled but console login is allowed***

### ***Terminal is disabled but root login is allowed***

---

These messages are associated with the super user logging onto the override terminal. Under the assumption that the console device (including a serial console) is a special device and considered a physical resource worth protecting, a lock on the super user account does not prevent the super user from logging into the console. This presents a means for entering the system even when all other accounts or terminals are locked. Before continuing, use the audit trail to investigate the reasons for the lock. A lock-out caused by unsuccessful login attempts on the system console is cleared automatically, but lock-outs due to other reasons remain in effect. The console, in effect, is never locked out for the super user.

## ***Audit: filesystem is getting full***

---

The audit subsystem may occasionally display this warning when the audit filesystem reaches a certain threshold. This warning message indicates that space is low on a particular device. If additional directories were specified to the subsystem, it automatically switches when the filesystem has reached the threshold value for remaining free space. Otherwise, the administrator must intervene to make more space available. If not, auditing is terminated when the threshold value is reached. The audit daemon program `auditd` may terminate for the same reason. If unable to write a compaction file because of insufficient space or an I/O error, `auditd` terminates. Use the `sysadmsh` System ⇨ Audit ⇨ Disable selection to terminate auditing if it has not already been done. Analyze the source of the problem and solve it before re-enabling auditing.

## ***Authentication database contains an inconsistency***

---

This message is displayed while running one of the programs associated with the TCB or a protected subsystem. The Authentication database integrity is in question. The Authentication database is a composite of the Protected Password database, the Terminal Control database, the File Control database, the Command Control database, the Protected Subsystem database, and the System Defaults file, and the message applies to all of these. Either a data entry is not present when expected, or the items within an entry are not correct. This message is intentionally vague. The invoking user is alerted to the problem but not given enough information about the cause to allow them to exploit an integrity problem within the security perimeter. The real reason for the problem may be found in the audit trail if Database Events were enabled for the user that generated the message. The Accounts ⇨ Check ⇨ Databases selection should help you determine the problem.

## ***Can't rewrite terminal control entry for tty***

### ***Authentication error; see Account Administrator***

---

The most likely reason for this problem is that the device tty entry in the `/etc/auth/system/ttys` file is corrupted. If a multiscreen entry in the `/etc/auth/system/ttys` file is corrupted, the multiscreens listed below the corrupted entry are inaccessible. For example, if the entry for `tty03` is corrupted, when you press `<Ctrl><Alt><F4>` or `<Ctrl><Alt><F5>`, the system displays a blank screen.

To solve this problem:

1. Log in on the override tty.
2. If the `/etc/auth/system/ttys` file does not contain an entry for `tty01`, add the following line below the console entry:

```
tty01:t_devname=tty01:chkent:
```

You can now enter multiuser mode and access the multiscreens of the console.

3. If the `/etc/auth/system/ttys` file appears intact, you should check and see if the `/etc/group` file was corrupted or removed. If so, restore the file from backups.

## ***Cannot obtain database information on this terminal***

---

This message indicates a problem with the `/etc/auth/system/ttys` file. Running the `tcback(ADM)` utility should fix any problems with the `ttys` file. Under normal conditions, `tcback` is run automatically whenever the system boots.

## ***Login incorrect***

---

The user entered an incorrect login name or login/dial-up password. If this happens repeatedly, you may need to alter the password to permit the user to log in again.

## ***login: resource Authorization name file could not be allocated due to: cannot open;***

---

The `/etc/auth/system/authorize` file was corrupted or removed. Log in on the override tty and restore the file from backups.

## ***Terminal is disabled – see Account Administrator***

---

The terminal is locked to all users. This is similar to account locking. Either an authentication administrator locked the terminal with `sysadmsh` or a number of incorrect login tries (to one or more accounts) passed the threshold for that terminal. In both cases, determine what happened and then use the `sysadmsh` to reset the lock.

## ***useshell: File Control database inconsistency***

---

This message occurs while attempting to add users with `sysadmsh` Accounts ⇨ User ⇨ Create ⇨ . This means there is a problem with the permissions or ownership of a database file located in `/usr/lib/mkuser`. You should run the `fixmog(ADM)` utility as follows:

**`fixmog -i`**

The `-i` option checks files interactively, asking you to confirm each change before it is done. There may be additional problems reported by `fixmog` that should be examined via an integrity report; see “`integrity(ADM)`” (page 887) for more information.

## ***useshell: Script path /usr/lib/mkuser/mkuser.init may be compromised ...***

---

This means there is a problem with the permissions or ownership of a database file located in `/usr/lib/mkuser`. You should run the `fixmog(ADM)` utility as described for the previous item.

## ***You do not have authorization to run ...***

---

The command is part of a protected subsystem. For that subsystem, the authentication administrator has not provided you with the kernel authorization needed to run this command and/or related commands. The authentication administrator uses the Accounts ⇨ User ⇨ Examine:Privileges selection to grant or deny such authorizations.

## ***Tape drive problems***

---

This section covers some common problems that you may have with your cartridge, floppy, or SCSI tape drive on your system. Note that these problems generally occur immediately after using `mkdev tape` to install your tape drive.

For more information about installing and configuring tape drives, see the *Hardware Configuration Guide*.

### ***Bad octal digit***

---

After you run **mkdev tape** to install a cartridge tape drive during the kernel relink process, the system displays the following message:

```
/etc/conf/pack.d/ct/space.c line 46 bad octal digit
```

To fix this problem, verify that the base address for the tape controller is entered with a leading "0x" rather than a trailing "H". Run **mkdev tape** again to modify the cartridge tape parameters.

### ***Tape not recognized at bootup***

---

If your system does not display a message like the following at boot time, the system does not recognize the tape controller card:

```
%tape 0x338-0x33C 05 1 type=wangtek
```

Instead of this hardware recognition message, the system displays a message like the following:

```
ct: ERROR: Tape controller (type=wangtek) not found
```

### ***Cartridge tape***

For a cartridge tape drive, check the following:

1. Verify that the tape controller card is physically configured to the base address that you gave when you ran **mkdev tape**. If necessary, run **mkdev tape** again and change the parameters. For information about configuring your cartridge tape drive, consult the *Hardware Configuration Guide*.
2. Verify that the tape controller is seated properly on the motherboard and, if necessary, insert the controller in a different slot.
3. Some tape drives (particularly external drives) require that the drive be attached to the tape controller and powered on at boot time.
4. Verify that the cartridge tape drive is supported. See the SCO list of compatible hardware for a list of the compatible tape drives.
5. If your system still does not recognize the tape controller card at boot time, see the documentation that came with your hardware.

### ***Irwin and QIC-40/80***

For an Irwin (mini cartridge) or QIC-40/80 tape drive, verify that the settings you supplied when you ran **mkdev tape** actually match the jumpers set on the back of the floppy tape drive.

## SCSI tape

For a SCSI tape drive, run `mkdev tape` to check the following:

1. Verify that the ID number for the controller of the device is correct. The ID number is determined by the jumper settings on the controller. The valid range is 0-7. SCSI tape drives are often configured for ID.
2. Make sure that the host adapter number is correct. The first SCSI host adapter of a given type is 0; the second is 1.
3. Check that the LUN (Logical Unit Number) is correct. In most cases, the controller is embedded in the same physical unit as the device and supports one device with LUN 0. If the controller is not embedded, it supports up to eight devices. If this is the case, the LUN is determined by the jumper settings on each device. The valid range is 0-7.
4. Verify that the host adapter itself is recognized at boot time.
5. Verify that the tape drive is supported. Refer to the SCO list of compatible hardware for a list of supported tape drives.

## Tape commands hang

---

If the tape drive hangs when you test it with the following command:

```
tape reset
```

Check the following:

1. Verify that the cartridge tape controller card is physically configured for the DMA value that you gave when you ran `mkdev tape`. To fix this, either physically reconfigure the tape controller DMA setting to agree with the address that you gave with `mkdev tape`, or run `mkdev tape` again and specify the DMA value on the tape controller. Do the same for the interrupt vector.
2. Verify that the cable between the tape controller and the tape drive is properly attached.
3. If the `tape reset` command still hangs, see the documentation that came with your hardware.

## Cannot open /dev/rct0 error message

---

If a `tar(C)` command or other commands which access the tape drive fail to write the contents of a directory to a tape and the system displays the following message:

```
Cannot open /dev/rct0
```

Check the following:

1. Use **hwconfig** to verify that the interrupt vector for a cartridge tape drive does not conflict with any other device on your system. Refer to “Tape not recognized at bootup” (page 980) for information on the available interrupt vectors. To change the interrupt vector, run **mkdev tape** and indicate a new interrupt vector value.
2. Verify that the actual physical interrupt (IRQ) setting on a cartridge tape controller card agrees with the interrupt vector that you specified with **mkdev tape**.
3. Make sure that the proper device for the tape drive is located in the */dev* directory. Enter:

```
1 /dev/rct0
```

The output should look like the following:

```
crw-rw-rw 1 root root 10, 0 Feb 14 12:00 rct0
```

(The major device number (10) may differ.) If the listing of */dev* does not contain a line similar to this, run **mkdev tape** again to create the device.

4. Verify that all cables are connected correctly.
5. If the **tar** command still does not work, the tape hardware may be broken. See the documentation that came with your hardware for more information.

## Terminal troubleshooting

---

This section explains how to solve the following problems with terminals:

- terminal is completely non-functional
- terminal is hung
- terminal display is scrambled
- terminal is locked
- terminal does not echo

### Restoring a non-functional terminal

---

A completely non-functional terminal displays no login prompt and does not respond to keyboard input. This situation is usually caused by hardware failure or configuration problems. Follow these steps to check a non-functioning terminal:

1. Check the brightness control on the terminal.
2. Check the power and communication connections at the terminal and computer.

3. If applicable, enter set-up mode on the terminal and verify the terminal configuration settings. The settings should include 9600 baud, 8 data bits, 1 stop bit, and no parity.
4. Enable the port to which the terminal is connected. For example, to enable `tty007`, use the following command:

```
enable tty007
```

5. Verify that there is a `getty` process associated with the terminal port. For example, enter:

```
ps -t tty007
```

6. Test the hardware communications by disabling the port and redirecting output to the non-functional terminal: for example, to test `tty007`, use the following commands:

```
disable /dev/tty007
echo hello > /dev /tty007
```

If these steps do not restore the non-functional terminal, check the documentation that came with the terminal hardware for troubleshooting suggestions.

## *Fixing a hung terminal*

---

A terminal is considered “hung” if the previous work session is still visible on the display, but it does not respond to keyboard input. Use the following steps to fix a hung terminal:

1. Wait a minimum of 60 seconds before trying to resurrect the terminal. (If the system is busy, the terminal may not respond immediately to keystrokes because the system response time has increased.)
2. Press `(Ctrl)q` to re-enable transmission in case the `(Ctrl)s` (transmit off) signal was inadvertently pressed.
3. Check to see that all power cords, keyboard cords, and communications cables are connected.
4. Reset the terminal hardware by recycling power to the terminal and then reinitialize it by running `tset(C)` with no arguments.
5. Verify the terminal set-up mode configuration settings (if available) as described in step 3 of the previous section.
6. Test the hardware communications by redirecting output from an operating terminal to the locked one as described in the step 6 of previous section.

7. Check the processes that are running on the locked terminal port with the following command:

```
ps -t ttyn
```

Stop the process that the user was running when the terminal hung using **kill(C)**. Refer to “Runaway processes, stopping” (page 974) for information on how to do this. If the program does not die, you must reboot the system to stop the process.

8. Determine whether the current line characteristic parameters are correct. For example, use the following command to display these values for **tty007**:

```
stty -a < /dev/tty007
```

You can also compare the **stty** settings with those of a working terminal.

9. Reset the serial line characteristics with the following command:

```
(Ctrl)j stty sane (Ctrl)j
```

If you cannot enter the command on the terminal, you can redirect the **stty** command from another terminal as follows:

```
stty sane < /dev/tty007
```

If the **ps -t** command shows only a **getty** program, the terminal should display a login prompt. If it does not, check the terminal hardware again.

### *Fixing a scrambled terminal display*

---

A scrambled terminal responds to keyboard input but the display is incorrect. Follow these steps to fix a scrambled terminal:

1. Check the terminal type (**TERM**) for the user with the **env** command. If the terminal type is incorrect, reset it. For example, to set the terminal type to “**wyse60**”, enter:

```
TERM=wyse60
```

After resetting the terminal type, reinitialize the terminal by entering **tset** with no arguments.

2. Reset serial line characteristics with the following command:

```
(Ctrl)j stty sane (Ctrl)j
```



## Unlocking a locked terminal

---

If a terminal has been locked by the system administrator to prevent logins on that terminal, or if the system locked the terminal because a user exceeded the number of unsuccessful logins attempts, the following message is displayed on that terminal:

Terminal is disabled -- see Account Administrator

To unlock the terminal, invoke `sysadmsh` and select:

Accounts ⇨ Terminals ⇨ Unlock

## Restoring a non-echoing terminal

---

A non-echoing terminal is a terminal that responds to keyboard input but does not display the characters entered at the keyboard. (This is different from a *locked* or non-functional terminal that does not respond to input at all; see “Unlocking a locked terminal” (this page) or “Restoring a non-functional terminal” (page 982) for a solution to these problems.)

Sometimes, when a program stops prematurely as a result of an error, or when the user presses the (Break) key, the terminal stops echoing. To restore the terminal to normal operation, enter the following:

`(Ctrl)j stty sane (Ctrl)j`

Enter this command accurately because the terminal does not display what you enter at the keyboard.

The terminal should now display keyboard input. If it does not, follow the steps outlined in “Restoring a non-functional terminal” (page 982).

## UUCP troubleshooting

---

This section discusses common problem situations that you may experience with UUCP and covers the following topics:

- how to check for a faulty ACU/modem
- error messages that `cu` displays
- common errors when running `uucp`
- how to check the status of a `uucp` request
- using `uutry` to debug UUCP communications
- common messages that appear in the UUCP log and status files
- how to check permissions settings on UUCP files

- checking that your sitename is unique
- what to check if UUCP is very slow
- what to do when `uucp` works, but `uux` does not
- a description of the UUCP troubleshooting utilities, `uuccheck`, `uulog`, `uuname`, `uustat`, and `uutry`

Before troubleshooting the UUCP system, make sure that the physical connection works. If you are connecting to UUCP with a modem, verify that the modem is installed and configured correctly.

**NOTE** If you are configuring an outbound UUCP connection over a modem, `cu` must function in order for UUCP to work.

### *Check for faulty ACU/Modem*

---

The following are two methods for checking whether the ACU (Automatic Call Unit) or modems are working correctly:

- To display a list of queued requests, the time of the last request attempt, and the contact status, enter:

```
uustat -q
```

- To use a specific line and print debugging information during the contact attempt, enter:

```
cu -x9 -ltyyn phone
```

Refer to “Modem problems” (page 959) for information on solving common modem problems.

## Errors when testing the connection with *cu*

---

This section describes messages that the system displays when testing the connection with the *cu* command fails.

### ***Connect failed: CANNOT ACCESS DEVICE***

If the connection fails and the system displays the “CANNOT ACCESS DEVICE” message, check the permissions on the device file. For example, to check the device file for *tty1a*, enter:

```
1 /dev/tty1a
```

The ownership and permissions settings should look like the following:

```
crw--w--w- 1 uucp uucp 5, 0 Feb 14 12:00 /dev/tty1a
```

If, instead, the ownership and permissions for the device file look like this:

```
crw----- 1 bin terminal 5, 0 Feb 14 12:00 /dev/tty1a
```

Verify that the line for the port in the */etc/inittab* looks similar to the following:

```
t1A:23:respawn:/etc/getty -t60 tty1A 3
```

If it does not, edit */etc/inittab* and */etc/conf/init.d/sio* and change the lines for the appropriate port.

**NOTE** Because each time the kernel is relinked, a new *inittab* file is created, to retain the changes in *inittab*, you must also edit */etc/conf/init.d/sio*.

To make your changes to *inittab* effective immediately, enter:

```
telinit q
```

### ***Connect failed: SYSTEM NOT IN Systems FILE***

If the */usr/lib/uucp/Systems* file on your computer does not contain an entry for the system that you are trying to access, *cu* displays the “SYSTEM NOT IN Systems FILE” error message.

**NOTE** This error message is often caused by an illegal character in the phone number, leading *cu* to treat it as a system name rather than a phone number.

To display a list of all the systems that your system is connected to, enter:

```
uuname
```

The system may display this error message if you used the *-l* option to specify a serial port and you entered the wrong line number. Verify that the line is configured properly.

### ***Connect failed, NO DEVICES AVAILABLE***

If `cu` fails to connect and displays the “NO DEVICES AVAILABLE” message, check to see that the `/usr/lib/uucp/Devices` file is set up correctly.

Verify that the line that corresponds to the device that you are using is uncommented. For example, the entry in *Devices* for a direct line using `ttyla` at 9600 baud looks like this:

```
Direct ttyla - 9600 direct
```

There should be no leading spaces. If the *Device* file looks correct, the remote line may be busy. Try again later.

### ***Connected, but no login prompt***

If `cu` displays the *Connected* message, but not the login prompt, the line for the remote system may be busy. To exit `cu`, enter `~.` and press (Return).

If everything appears to be working on your system but the login prompt for the remote machine does not appear, check with the remote system administrator to verify that the `getty` on the remote system is set up with the same communications parameters that you are using.

### ***Connected, but screen displays garbage***

If you connect to the remote system, but your screen displays garbage, see “Modem answers, but terminal displays garbage” (page 961) for more information. You should confirm that the communication settings on your system match those of the remote system. If the condition persists, the connection may be bad. Exit `cu` by entering `~.` and try again later.

## ***UUCP failed messages***

---

Errors that UUCP displays immediately after you enter the `uucp` command are generally syntax or permissions errors, and include either of the following messages:

```
uucp failed completely
uucp failed partially
```

The following are some common problems and the error messages that the system displays:

- UUCP displays the following message when the permissions are set so that `uucp` cannot access the source file:

```
can't read file (filename)
uucp failed partially
```

Change the permissions on the source file to allow all users read permissions.

- The following message is displayed when you do not enter both a *source* and *destination* with the **uucp** command:

```
usage uucp from ... to
uucp failed completely
```

- If you enter a *sitename* that is not in the */usr/lib/uucp/Systems* file, UUCP displays:

```
bad system name: sitename
uucp failed completely
```

Use **uname** to display a list of the sites to which your system is connected.

## Checking the status of a uucp request

---

If the system does not display an error message immediately after entering the **uucp** command, and the system prompt appears, the **uucp** request is queued. Use the **uustat** and **uulog** utilities to check the status of your **uucp** job.

Use the **uustat** command to display the status of the currently queued **uucp** requests or display the status of connections to other systems. For example, to display a list of all the queued jobs for the user *robertm*, enter:

```
uustat -u robertm
```

The **uustat** utility displays user status information in the following format:

```
couscousN266 01/26-15:43 S couscous robertm rmail edwarda
```

To list the status of the accessibility of all the remote machines, use the following command:

```
uustat -m
```

The following example shows the output from the **uustat** command:

```
scooter 01/26-15:43 CAN'T ACCESS DEVICE
disco 01/27-12:01 SUCCESSFUL
obie 01/25-05:12 CALL FAILED, JOB DELETED
```

For more information on the options to **uustat**, see the **uustat(C)** manual page.

The **uulog** command displays the status messages (most recent last) that the UUCP programs write to the files *uucp/sys*, *uux/sys*, *uucico/sys*, and *uuxqt/sys*, in the */usr/spool/uucp/Log* directory (*sys* is the name of the system that **uucp** is trying to access).

For example, to display status information about file transfers involving the system *scooter*, enter the following:

```
uulog -s scooter
```

The **uulog** utility displays information about the system in the following format:

```
uucp scooter (01/26-15:43:00, 304, 5) COPY (SUCCEEDED)
```

**NOTE** The **uulog** command displays all the status messages that **uucp** logs in the *.Log* files; the messages are not necessarily error messages.

### *Debugging UUCP communications*

---

If you are unable to contact a particular machine, use the following procedure to debug the communications to that machine with **uucp** and **uutry**.

1. Log in as *root*.
2. Enter the following commands, pressing **<Return>** after each line:

```
uucp -r testfile machine! /usr/tmp/testfile
/usr/lib/uucp/uutry -r machine
```

where *testfile* is a file that you want to transfer and *machine* is the sitename of the machine that you are trying to contact. The **-r** option overrides the retry time specified in */usr/lib/uucp/Systems*.

**NOTE** If you are using **cs**, you must escape the "!" (**\!**). Use of **cs** for the *root* account is not supported.

3. When you see one of the following messages, press the **<Del>** key:

```
Conversation Complete: Status SUCCEEDED
Conversation Complete: Status FAILED
```

If your system displays the first message, check the remote system, *machine* for the file */usr/tmp/testfile*. If the file exists and is not empty, the UUCP connection functions. If the system displays the second message, the connection transfer failed.

If the UUCP system is set up correctly, the system attempts to transfer the *testfile* file to the remote system, *machine*. The **-r** option to **uucp** queues the file without starting the transfer. This allows **uutry** to invoke **uucico** with debugging.

**NOTE** The **uutry** command displays more debugging information if you run it as *root*.

The **uutry** command saves the debugging output to the file */tmp/machine* on the local machine and prints the debugging output to your terminal using **tail -f**. Copy the output from */tmp/machine* if you wish to save it. You can change the debugging level from the default level (5) using the **-x** option.

If the system displays the “Status FAILED” message, you can look through this file for status and error messages.

See the section on UUCP error messages in the chapter on building a remote network with UUCP in *Administering Networking Services* for an explanation of some common UUCP messages.

If you still cannot solve the problem, you may need to call support personnel. Save the debugging output for diagnosing the problem.

## ***Alarms in UUCP audit output, data is not transferring***

---

When sending data, UUCP expects a certain packet size and then a checksum on the packet. If it does not receive a full packet in a certain amount of time, a timeout occurs, which generates an alarm. A problem can sometimes occur once UUCP has begun sending data, in that part way through the transfer, alarms are generated and UUCP attempts to resend the packets. There are a couple of possible reasons for this problem.

First, check that **mapchan** is not being used on the line that **uucico** is using. If it is, enter:

```
mapchan -n ttyxx
```

to turn mapping off for that port. If **mapchan** is not running on that port, there is probably some kind of line noise or modem problem that is corrupting the data being sent.

Alternatively, the problem may occur if you are using UUCP with MNP 5 or some other error checking protocol that is specific to the modems. Error checking on a modem also sends a certain packet size with a checksum. The problem is that the sending modem waits to get a certain amount of data before it sends the packet, which can cause problems with UUCP because it is also waiting for a packet. The solution is to not use error checking on the modem when using UUCP because UUCP has its own error checking.

## ***UUCP log and status file messages***

---

UUCP keeps track of activity in log and status files in the */usr/spool/uucp* directory. The UUCP programs **uucp**, **uux**, **uucico**, and **uuxqt** write status information for each system to files in the */usr/spool/uucp/Log* directories. The **uucico** utility writes messages to the files in the */usr/spool/uucp/Status* directory. These messages describe the status of the transfer request, such as **uucico** failures, completions, and the time until the next allowable call for each remote system with which you communicate.

The following sections describe common error messages that UUCP programs write to these files.

### ***DEVICE LOCKED***

The `uucico` creates a lock file named `LCK.sitename` for the remote system in the `/usr/spool/uucp` directory. If a file for the system that you are trying to call exists, `uucico` thinks that the device is in use.

To use the device, enter the following command on the remote system, substituting the name of the remote system for *sitename*:

```
rm /usr/spool/uucp/LCK.sitename
```

**NOTE** If this does not solve the problem, the modem could be asserting CD improperly.

### ***LOGIN FAILED***

If this error message appears, one of several things may be wrong:

1. Check the `/usr/lib/uucp/Systems` file on the local machine to verify that the information in this file, particularly the chat script, is current. Some information that may be out-of-date is the phone number, login, and password.
2. Verify that the modem setup on both ends is correct. Refer to “Modem problems” (page 959) for more information.
3. Check the phone connection; try a different phone line.

After each step, invoke the `uutry` command.

### ***NO DEVICES AVAILABLE***

If the system displays this message, one of several things may be wrong:

1. The system thinks that the modem is in use because there is a lock file for it. Enter the following from the local machine:

```
rm /usr/spool/uucp/LCK*
```
2. There may be no valid device for the calling system to use. Verify that the device named in the `/usr/lib/uucp/Systems` file corresponds to the entry in `/usr/lib/uucp/Devices`.
3. If the message persists, reboot the system. Enter the `rm` command again, and then invoke the `uutry` command again.
4. If the message persists after rebooting the system, the modem may be configured incorrectly. Verify that you can use `cu` to call out. Refer to “Modem problems” (page 959) for more information.



**REMOTE DOES NOT KNOW ME**

If the system displays this message, the `/usr/lib/uucp/Systems` file on the machine that you are trying to call does not contain an entry for your machine. Check the *Systems* file on the remote machine.

**REMOTE HAS a LCK FILE FOR ME**

If your system displays this message, either the remote system is trying to call your system or there is a lock file for your system on the remote system.

1. Remove the lock file from the remote system by removing the appropriate lock file:

```
rm /usr/spool/uucp/LCK.sitename
```

2. Invoke the `uutry` command again.

**SYSTEM NOT IN Systems FILE**

If the `/usr/lib/uucp/Systems` file on your system does not contain an entry for the system that you are trying to access, this error message is displayed. Verify that the system name is in the *Systems* file using the `uname` command.

**RETRY TIME NOT REACHED**

When UUCP requests fail, retries are not executed immediately. After an attempt to contact a remote system, a status file remains in `/usr/spool/uucp/.Status/nodename` (*nodename* is the name of the remote system that you are trying to reach). This file contains information about the last request and does not allow another request until the minimum retry period (specified in the `/usr/lib/uucp/Systems` file) is reached. If you try to use UUCP again, the system displays an error message like the following:

```
RETRY TIME NOT REACHED
```

To enable another call attempt immediately, remove the status file for the remote system from the `/usr/spool/uucp/.Status` directory:

```
rm /usr/spool/uucp/.Status/nodename
```

**CANNOT ACCESS FILE**

If the system displays this message, it cannot access the calling device port. Check the permissions and ownership on the device file:

```
l /dev/ttyxx
```

Where *xx* is the tty number. The ownership and permissions settings should look like the following:

```
crw-r--r-- 1 uucp uucp 5, 0 Feb 14 12:00 /dev/ttyxx
```

See “Connect failed: CANNOT ACCESS DEVICE” (page 987) for information on changing the permissions.

## WRONG TIME TO CALL

The `/usr/lib/uucp/Systems` file may restrict outgoing calls at this time.

### *Checking UUCP files and permissions settings*

---

UUCP does not work correctly if it cannot read or execute its files. Because virtually all of the UUCP files are writable only by the super user, and many of them are also readable and executable only by `root` and `uucp`, you should log in as `root` to install and modify the UUCP system. When you have finished, verify that all of the UUCP files are owned by `uucp` and not `root`.

To check the permissions of the UUCP files, use the following commands:

```
cd /
fixperm -n -v -dUUCP /etc/perms/*
```

This `fixperm` command displays a list of any UUCP distribution files with incorrect permissions.

Use the `uucheck(ADM)` command to check for the presence of the required files and directories.

**NOTE** The *Systems* and *Permissions* files contain unencrypted passwords, and therefore should be readable only by `uucp` (and `root`).

### *Verifying that sitename is unique*

---

Make sure that the first seven characters of your sitename name are unique. If your machine is connected directly to a machine with the same sitename, UUCP does not allow communication with that machine. If both your machine and another machine with the same name are not directly connected, but are on the same UUCP network, electronic mail may go to the wrong machine.

To check the sitename use the `uname(C)` command as follows:

```
uname
```

To change the sitename, see the chapter on setting up electronic mail in *Administering Networking Services*.

### *UUCP truncates system names to seven characters*

---

By default, UUCP truncates system names to seven characters to ensure compatibility with all versions of UUCP. Although disabling this feature is not recommended, it can be done as described in this section.

In the file `/usr/lib/uucp/Permissions`, add the variable `MYNAME=name` to each entry in this file, where *name* is the actual system name that contains more than seven characters. Also, add a new entry in `/usr/lib/uucp/Permissions` as follows:

```
MACHINE=OTHER \
READ=/usr/spool/uucppublic:/usr/tmp \
WRITE=/usr/spool/uucppublic:/usr/tmp \
COMMANDS=rmail:rnews:uucp \
SENDFILES=yes REQUEST=yes \
MYNAME=name
```

where *name* is the actual system name.

The `MYNAME` variable supersedes the truncated name. This procedure should be done on every machine in the UUCP network that has a name greater than seven characters long.

**NOTE** `MYNAME` affects only those systems calling into the system whose `Permissions` file has been modified as described here. Calling out to a system with a system name greater than 7 characters results in that system name being truncated to 7 characters. This is only a problem if your system calls more than one system whose names are not unique to 7 characters.

## *UUCP is abnormally slow*

Because UUCP is a batch network, requests are not executed immediately. Therefore, when users report that UUCP does not work, the problem may be that the system is slow. If the system is slower than usual, check for the following problems:

1. Check the spool directories, `/usr/spool/uucppublic` and `/usr/spool/uucp` for an overload of old work files and remove them.
2. Use `ps(C)` to verify that there are not too many `uucico` processes going at once. The limit to the number of communications processes is specified in `/usr/lib/uucp/Maxuuscheds` and `/usr/lib/uucp/Maxuuxqts`. Check to see that the limit is not too high; the default is 1.
3. Make sure that the filesystem that contains the spool directory is not out of space. Refer to “Out of space on filesystem” (page 945) for more information.
4. The `/usr/lib/uucp/uudemon.clean` script performs general cleanup by removing old files that are trying unsuccessfully to execute. To change the frequency of the cleanings, edit the `find` statement in the `uudemon.clean` script.

## *uucp works, but uux does not*

---

If you can use **uucp** to transfer files between two systems, but you cannot use **uux**, there is a problem with the `/usr/lib/uucp/Permissions` file. When you use the **uucp** utility, the remote system requires only the **LOGNAME** entry in *Permissions*; **uux** also requires the **MACHINE** entry.

To fix this problem, add the **MACHINE** entry, with the name of the remote system, to *Permissions*. For example, if your local machine, *goanna* is set up to call *obie*, the entry for *goanna* in the *Permissions* file on *obie* should look like this:

```
LOGNAME=uugoanna MACHINE=goanna \
COMMANDS=ALL \
READ=/ \
WRITE=/ \
SENDFILES=yes REQUEST=yes
```

**NOTE** The permissions granted in the example above are very liberal and should only be used in closely coupled systems where there is no security risk.

## *UUCP troubleshooting utilities*

---

Table 5-1 lists several commands you can use to check for basic communications information.

**Table 5-1** UUCP troubleshooting tools

| Command | Description                                                                                                                                                                                                                                   |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| uucheck | Allows you to check for the presence of files and directories required by <b>uucp</b> . This command also checks the <i>Permissions</i> file for obvious errors.                                                                              |
| uulog   | Displays the contents of the log directories for specific hosts.                                                                                                                                                                              |
| uuname  | Lists the machines that you are set up to contact.                                                                                                                                                                                            |
| uustat  | Display the status of the currently queued <b>uucp</b> requests or connections to other systems.                                                                                                                                              |
| uutry   | Invokes <b>uucico</b> with debugging, saves the information to the file <code>/tmp/machine</code> , and directs the last 10 lines of the output to the terminal. The <code>-x</code> option changes the debugging level (default is level 5). |

# Index

---

## Special characters

- & (ampersand), 622
- \* (asterisk), 623
- \ (backslash), 624
- + (plus) symbol, Automounter maps, 625
- / (slash), root filesystem, 95
- /- (slash hyphen), 614
- " (double quotes), 624

## A

- A: drive, 217
- Aborting, DOS processes, 349
- Absolute pathnames, 171
- accept(ADM), 142
- Access privileges, 417, 429
- Access time, locating files by, 103
- Accessing, DOS partition, 288
- Account
  - administering, network, 419
  - copying from other systems, 80
  - DOS Services, 206
  - duplication on other machines, 79, 80
  - enabling, 976
  - ftp, 417
  - locking, 57, 68, 89, 956
  - managing, 548
  - modifying, 64-84
  - moving, 78
  - password, 956
  - sysadmsh(ADM) options, 38
  - transitions with su(C), 78
  - user, 956
- "Account is disabled -- see Account Administrator", 956, 976
- "Account is disabled but console login is allowed", 976-977
- Acer Fast Filesystem (AFS)
  - NMPBUF parameter, 914
  - NMPHEADBUF parameter, 914
  - repairing with fsdb(ADM), 954
- Active connections display, 435
- ACU dialer type, 683-684

## Adding

- administrator, 66
- clients, 640
- computer, 14
- login shell, 94
- media defaults, 174
- NIS servers, 592
- printer, 119, 214
- tape drive, 36
- user, 57, 60-64, 548, 569, 588, 589, 590

## Address

- mail, incorrect, 958
- mask, 534, 535
- parsing rules, 797
- resource records, 470

## addxusers(ADM), 80

## Administering

- client, 635
- utilities, 645

## Administrative tasks, LAN Manager client, 635

## AF\_INET, 692

## AGEINTERVAL parameter, 917

## Agents, 495

## "aio\_breakup ... buffer table overflow", 872

## "aio\_breakup ... request table overflow", 872

## "aio\_memlock ... process table overflow", 872

## "aio\_setlockauth ... lock table overflow", 872

## Alerting

- mount print wheel, 127-129
- printer faults, 129-131

## Alias

- database, 817, 821, 822
- file, 747-750, 781-782
- forms, 820
- table, 655

## Aliasing, 794

## ALL value, COMMANDS option, 680

## "allocreg - Region table overflow", 872, 916

## Allow list for printer forms, 132

## Alt key, 939

## Anonymous account, 417, 430

## ap(ADM), 80

## Apparently-To header line, 822

Append mode, 562

## Applications

DOS, configuring, 245

DOS, installing, 239, 240, 241, 242, 246, 247, 248, 249, 250, 251

DOS, removing, 252

DOS Services, installing, 239, 240

public, 242

running in background, 340

shared by DOS and UNIX systems, 242, 245-248

sysadmsh(ADM) selection, 36

tailoring, 268, 269

## Archiving, 167

directories, 170

floppy disk, 173, 175

sysadmsh(ADM) options, 40, 41

tape, 170

ArpaNet. *See* Internet

ASCII characters, 353

asroot(ADM), 77, 78

ASSERT error messages, 705

Association modes, 533

Asynchronous, event calls, 655

Asynchronous I/O (AIO), 928

"at: you are not authorized to use at. . .", 942

at(C)

sysadmsh(ADM) option, 41

troubleshooting, 941-942

using with DOS processes, 340

Atomic write, 562

AT&T 5310/20 matrix printer, 122

## Attaching

COM port, to DOS, 286

devices to DOS Services, 228-233

printer port, to DOS, 287

printers to DOS, 215

ATTRIB command, DOS, 222

Attributes, printer, 133-134

## Audit

administration, 93

authorizations, 91

subsystem, 93, 977

sysadmsh(ADM) options, 36, 37, 75-76

user accounts, 75-76

"Audit: filesystem is getting full", 977

audittrail authorization, 92

auth authorization, 91

authcap(F), file format, 58

authck(ADM), 50, 58, 61, 596, 866

"Authentication database contains an inconsistency", 977

"Authentication error; see Account Administrator", 978

AUTHLOG, 784

## Authorization

assigning, 38, 74-75

changing, 90

default, 81, 83

kernel, 91-93

mount filesystem, 98

primary, 92

problems, 979

secondary, 92

shutdown, 92

su, 78

subsystem, 91-92

sysadmsh(ADM) options, 40

authsh(ADM), 67, 78

Auto-answer modem configuration, 833

autoboot(ADM), 55

AUTOEXEC.BAT file, 214

CHCP command, 316

DEVICE command, 310

interpretation under DOS Services, 265, 270, 298

KEYB command, 316

MODE command, 310, 316, 318

NLSFUNC command, 310, 316, 318

running, 298

use with NLS, 310, 312, 313, 316, 317, 318

Autologic APS-5 phototypesetter, 122

auto.master, *See also* Maps, Automounter, master

Automatic Call Unit (ACU), 662, 672, 683-685, 831, 986

Automatic startup, 641

Automatic Write Remapping (AWRE), 943

## Automounter

restarting, 627

configuring, 612, 613

debugging, 630

defined, 607

multiple servers, 620

planning, 612

starting, 612, 625

using with NFS, 607, 612, 613

using with NIS, 611, 612, 617, 626, 633

automount(NADM), 607, 613, 626

Autonomous system, 483

# B

**B:** drive, 217  
 Background UUCP daemons, 662  
 Backspace key, 12  
 BACKUP command, DOS, 217  
 Backups
 

- audit files, 107
- authorization, 91
- creating, 947
- DOS Services, 217
- restoring, 37, 947
- shared filesystem, 217
- sysadmsh(ADM), 37
- troubleshooting, 861

**Bad**

- block, 116
  - octal digit, 980
  - system name, 989

**Bad track**

- boot problems, 931
- mapping with badtrk(ADM), 942

**badhosts** channel, 716, 758  
**badusers** channel, 716, 758  
**Base address**, tape drive conflict, 980  
**Batch files**, use with merge command, 258, 259  
**batch(C)**, 941-942
 

- using with DOS processes, 340

**Baud rate**

- error, 666
- modem, 831
- modem problems, 960
- printer, 153-154, 159
- setting in /etc/inittab file, 964

**bcheckrc(ADM)**, 937  
**bdev**, fileysys(F) entry, 552  
**bdfush** daemon, 910  
**BDFLUSHR** parameter, 910  
**Berkeley Internet Name Domain (BIND)**. *See* Name server  
**BFREEMIN** parameter, 914  
**bind**

- NIS, 633
- table, 655

**BIND**. *See* Name server  
**Binding**, NIS. *See* ypbind(NADM).  
**BINDX** parameter, 656  
 /bin/login file, restoring, 935

"/bin/su: cannot setgid to auth, no auth entry", 956  
 /bin/sulogin file, restoring, 937  
**biod(NADM)**, 548, 555, 556, 561  
**BITNET**, 464  
**Block**

- bad, 116
  - defined, 101
  - device activity, sar -d, 893
  - displaying usage, 102
  - duplicate, 116
  - free-block list, 117
  - structure, 113

**Boot**

- automatic, 55
- configuring, 54-55
- display, 53, 54
- file, name server, 464, 473, 474
- filesystems, 36, 97
- load extension, kernel parameters, 929
- reboot system, 37
- restoring files, 51
- security, 866
- steps, 43-45, 47
- troubleshooting, 931-938
- using old kernel, 54

**Boot floppy**, 932
 

- Emergency Boot Floppy Set, 933-934, 938
- restoring root filesystem, 948

 "/boot not found", 932  
**Boot screen**

- port does not respond, 970
- tape drive recognition, 980

**boot(HW)**, 54  
**Booting**

- from a virtual diskette, dosboot command, 228
- games, from diskette, 296
- virtual machine, 297

**Bootstrap program**, 44, 54  
**Bootstring** options, 55  
**Bourne shell**, 339
 

- in an DOS Services NLS environment, 314

**Brackets**, in DOS Services commands, 281  
**Break**, login script, 674  
**Broadcast address**, configuration, 414  
**broadcastclient** mode, 533  
**Broadcasts**, 636, 637

Buffer, 113, 905  
  activity, sar -b, 891  
  allocating character list, 911  
  allocating physical I/O buffers, 908  
  buffer cache, 909  
  data, 648  
  effective size, 650  
  increasing cache hit rate, 908  
  network, 650, 652, 656  
  parameters, 908  
  size, 650  
  specifying age for filesystem updates, 909  
  streams, 651, 652  
  tape size, 909  
  transfer, 909  
  writing to disk, 910  
BUFFERS command, DOS, 235  
bufsz column, 650

## C

C shell, in an DOS Services NLS environment, 314  
C. (work) files, 664, 702-706  
C2 security, 14, 83  
Cabling  
  planning, 15  
  unterminated wire problem, 829  
Cache, initialization, 466  
Cache hits, reducing disk accesses, 908  
Caching-only server, 474  
CALLBACK option, Permissions file, 679  
Call-out table, 918  
Canadian French code page, 309  
Cancelling print request, 137  
"CANNOT ACCESS DEVICE", 987  
"Cannot create", 969  
"Cannot load floating point emulator", 934  
"Cannot obtain database information on this terminal", 955, 978  
"cannot open: device busy", 826  
"Cannot open /dev/rct0", 981  
"Cannot open /etc/inittab", 936  
"Cannot read emulator file header", 934  
Canonical, domain name, 471  
"can't change mode of job", 942  
"Can't rewrite terminal control entry for tty", 956, 978

Carrier Detection (CD)  
  light, modem problems, 961  
  line, modem configuration, 833  
Cartridge tape. *See* Backups  
cat(C), command, 292  
CGA  
  DOS Services, 209  
  option for specifying, 286  
Chain, netconfig, 455  
Changing  
  DOS memory allocation, 269  
  DOS options, with dosadmin command, 265  
  DOS partition protection, 222  
  NIS master server, 594  
  password, 57  
  user ID, 419  
Channel, 711, 757  
Channel files, 753  
Character buffers parameters, 911  
Character set, 306, 307, 310  
  ISO 8859-1, 310  
  printer, 125  
Characters  
  ASCII, 353, 375  
  ASCII vs PC, 353, 360  
  ISO vs PC, 360, 362  
  PC, 352, 353, 375  
charconv command, DOS Services, 306, 312, 321, 322, 323, 324  
Chat script. *See* Login, script  
CHCP command, DOS, 306, 310, 314, 315, 316, 321  
checkaddr(ADM), 770, 958  
Checking  
  auth database, message, 47  
  filesystems, 112  
  NetBIOS status, 643  
  network, 661  
  protected subsystems database, message, 47  
  tcb, message, 47  
  ttys database, message, 47  
checkque(ADM), 770  
checkup(ADM), 769  
CHKDSK command, DOS, 332  
chmod(C)  
  changing the protection of DOS partitions, 222  
  configuring DOS applications, 248



chmod(C) (*continued*)  
     finding files by permissions, 104  
     removing sticky bit, 955  
     setting directory SGID, 99  
     setting permissions, 212, 243  
 chmoodsugid  
     at(C), 942  
     kernel authorization, 92  
 chown authorization, 92  
 chroot(ADM), 430  
 Class  
     C and F defined, 797  
     printer, 120, 138-139, 141  
 Cleaning, filesystem, 44  
 "CLEANING NON SYSTEM3 FILESYSTEM", 948  
 cleanque(ADM), 771  
 cleantmp(ADM), 100  
 Cleartext, 60  
 Client  
     adding and removing, 640  
     administering utilities, 645  
     automatic startup, 641  
     configuring, 640, 645  
     defined, 402  
     /etc/systemid, 640  
     managing, 642  
     mode, 521, 533  
     name, 640  
     NFS, 551, 553, 560  
     open file processes, 642  
     operating parameters, 641  
     performance, 647  
     stopping, 641  
     supported servers, 635  
     table structures, 656  
     trusted access, 640  
 Clock  
     NFS problems, 563  
     parameters, 918  
     setting at startup, 46  
     synchronizing, 517, 523  
 Clock interrupt rate, setting with merge  
     fastclk option, 254  
 clock.txt, 532  
 CMASK parameter, 913  
 CNCBS, mtune(F) parameter, 655  
 Code page, 306, 308, 309, 310, 312, 315, 316,  
     317, 318, 319 (*continued*)  
     multilingual, 309  
     Nordic, 309  
     Portuguese, 309  
     purpose of, 308  
     See also DOS code page, 306  
     setting up in DOS, 312  
     software, 309  
     supported, 309  
     switching, 310  
 Code set. See Code page  
 CODEPAGE environment variable, UNIX, 314,  
     317, 319, 322  
 Colors, under sysadmsh(ADM), 34  
 COM domain name, 722  
 COM ports  
     attaching to DOS, 286  
     cabling, 829  
     dialing in, 834  
     DOS Services, 212, 232  
     interrupts, 832  
     modem use, 829  
 Command line  
     flags, 810  
     starting Automounter, 612, 613, 626  
 COMMAND.COM, passing DOS commands  
     to, from UNIX shell, 293  
 Commands, 12  
     default remote, 680  
     DOS See DOS commands, 251  
     executing remote, 659  
     networking, 410  
     ntpq, 541  
     printer, 134, 135  
     remote printing, 510  
     shared by DOS and UNIX system, 240  
     UNIX, 661  
     UUCP, 661, 662  
     xntpd, 541  
 COMMANDS option, Permissions file, 680  
 Communications protocols, 408  
 Community name, 495  
 Compacting files, 41  
 Comparing files, 41  
 Compatibility, network, 419  
 compress(C), 101  
 compressed, printer type, 133, 158  
 Computer, setting up, 14, 15  
 Conditionals, 799

Code page, 306, 308, 309, 310, 312, 315, 316,  
     317, 318, 319 (*continued*)  
     multilingual, 309  
     Nordic, 309  
     Portuguese, 309  
     purpose of, 308  
     See also DOS code page, 306  
     setting up in DOS, 312  
     software, 309  
     supported, 309  
     switching, 310  
 Code set. See Code page  
 CODEPAGE environment variable, UNIX, 314,  
     317, 319, 322  
 Colors, under sysadmsh(ADM), 34  
 COM domain name, 722  
 COM ports  
     attaching to DOS, 286  
     cabling, 829  
     dialing in, 834  
     DOS Services, 212, 232  
     interrupts, 832  
     modem use, 829  
 Command line  
     flags, 810  
     starting Automounter, 612, 613, 626  
 COMMAND.COM, passing DOS commands  
     to, from UNIX shell, 293  
 Commands, 12  
     default remote, 680  
     DOS See DOS commands, 251  
     executing remote, 659  
     networking, 410  
     ntpq, 541  
     printer, 134, 135  
     remote printing, 510  
     shared by DOS and UNIX system, 240  
     UNIX, 661  
     UUCP, 661, 662  
     xntpd, 541  
 COMMANDS option, Permissions file, 680  
 Communications protocols, 408  
 Community name, 495  
 Compacting files, 41  
 Comparing files, 41  
 Compatibility, network, 419  
 compress(C), 101  
 compressed, printer type, 133, 158  
 Computer, setting up, 14, 15  
 Conditionals, 799

"CONFIG: aio\_breakup - AIO buffer table overflow ...", 928  
 "CONFIG: aio\_breakup - AIO request table overflow ...", 928  
 "CONFIG: aio\_memlock - AIO process table overflow ...", 928  
 "CONFIG: aio\_setlockauth - AIO lock table overflow ...", 929  
 configaudit authorization, 92  
 CONFIG.SYS file  
     buffer keystrokes, 333  
     changing special commands, 235  
     command for EMS, 331  
     COUNTRY command, 310, 316  
     DEVICE command, 316, 318  
     how DOS Services interprets, 266, 269, 294  
     special commands, 235  
     specifying alternate, 295  
     use with NLS, 310, 312, 313, 316, 317, 318  
 Configuration file  
     *See also* CONFIG.SYS file  
     Internet, 428  
     UUCP, 662  
 Configuration file, sendmail  
     building from scratch, 804  
     defined, 796  
     semantics, 799  
     sendmail program, 820  
     special header lines, 822  
     syntax, 797  
     trying a different, 814  
 Configuration menu, dosadmin  
     explanation of fields, 267, 268  
     selecting, 267  
 Configuration options, sendmail, 808  
 configure(ADM), 658, 882-883  
 Configuring  
     Automounter, 613  
     broadcast address, 414  
     client, 640, 645  
     command, 652  
     domain names, 412  
     DOS applications, 207, 247, 248  
     DOS applications, with dosadmin  
         command, 239, 240, 245, 246, 247, 249  
     DOS characteristics, 263, 264, 265  
     gated, 485  
     interrupt vectors, 411  
     I/O base address, 411  
     IP address, 413, 414

Configuring (*continued*)  
     modem, 826  
     netmask, 415  
     network, 411, 427  
     NFS, 549  
     NIS, 573  
     NTP hosts, 534  
     PPP, 454, 455  
     print service, 150-152, 156-163, 213  
     printer, 120-122, 147-148, 153-156, 165  
     printer, with sysadmsh(ADM), 213  
     pseudo ttys, 432, 433  
     RAM buffer size, 412  
     remote printer, 513  
     serial line drivers, 415  
     SLIP, 442, 444  
     SNMP, 497, 498  
     system name, 411  
     TCP/IP, 410, 411, 412, 413, 414, 415  
     user accounts, 64-76, 81, 83  
     UUCP, 660, 667-671  
 "Connect failed: NO DEVICES AVAILABLE", 960  
 Connecting  
     modem, 665, 666, 827  
     printer, 125, 164  
     systems by wire, 695-696  
     UUCP, 663  
 Console  
     DOS Services, 206, 207  
     multiscreens, 978  
     troubleshooting, 939-941, 955  
 conspw file, 640, 641  
 constable file, 646, 647, 648, 649  
 Content types, 123  
 Context indicator (sysadmsh), 19  
 Control  
     codes, 349-350  
     map size, specifying, 920  
     Register 3 (CR3), 926  
 Control keys, 12  
     q, unlock keyboard, 939  
     s, stop scrolling, 939  
 controlkey statement, 542  
 Converting, XENIX to MMDF, 763  
 CONVOPTS environment variable, UNIX, 324  
 Coordinated Universal Time, 517, 521  
 COPY command, DOS, 299  
 Copy protection, 217, 219, 248, 249

- Copying
  - file, 41
  - floppy disks, 174-175
- Copy-only server, NIS, 567
- Core file
  - creating, 13
  - recursive removal, 105
- Corruption
  - filesystem, 111
  - repairing, 116
- Country code, 316
- COUNTRY command, DOS, 306, 310, 314, 315, 316
- COUNTRY environment variable, UNIX, 314, 317, 319
- .cpi file, 309, 318
- cps(ADM), 50
- CPU usage, 889, 897
- Crash recovery, 866
- crash(ADM)
  - analyzing core image, 883
  - reading putbuf buffer, 910
  - STREAMS display headings (table), 885
  - studying memory dump, 860
- Creating
  - dial-in accounts, 667
  - dosenv.def file, personal, 273
  - Emergency Boot Floppy Set, 932
- crlmap printer, 122
- cron command, using with DOS processes, 340
- "cron may not be running", 941
- cron(C)
  - authorization, 91
  - clearing log files, 107
  - daemon, 941
  - reducing system load, 888
  - troubleshooting, 941-942
  - using with NIS, 584
  - UUCP, 661, 663, 701
- crontab(C) directory, 890, 898
- "ct: ERROR: Tape controller ... not found", 980
- CTBUFSIZE parameter, 909
- Ctrl-Alt key combination, 348
- Ctrl-Alt-Del keystroke, 348
- Ctrl-C, effects on printing, 299
- "cu: dir permission denied", 831

- cu(C)
  - dialing out, 833, 835
  - login sequence, 674
  - modem connection errors, 987-988
  - print debugging information, 986
  - testing a modem, 831
  - troubleshooting modem, 959
- Current logins, sysadmsh(ADM) selection, 35
- custom(ADM)
  - adding client, 640
  - removing client, 640
  - restoring files, 50

## D

- D. (data) files, 702, 704, 706
- Daemons
  - Automounter, 612, 634
  - gateway, 483, 484
  - mode, 813
  - NFS, 548, 555, 559
  - NIS, 568
  - printing, 151
  - remote printing, 509
  - routed, 485
  - server, 428
  - SNMP, 496
  - timed, 517
  - UUCP, 661-664, 681-682
- Daisy wheel, 139
- DARPA
  - See also* Internet
  - defined, 711
  - setting up, 464
- DASI terminals, 122
- DAT tape drive size, 170
- Data
  - bit settings, modem problems, 961, 964
  - block, 113
  - blocks in filesystem exhausted, 945
  - compression, modems, 963
  - region, 916
  - storage, 112-113
- Data Communications Equipment (DCE), modem configuration, 829
- Data Terminal Equipment (DTE), computer serial configuration, 829

Data Terminal Ready (DTR), 828  
  auto-answer modem, 833  
  modem problems, 962  
Database, network, 429  
Databases, account, configuring, 59-60  
Datagram, Internet Protocol, 408  
Date, setting, 46  
dbmbuild(ADM), MMDF, 769  
dd(C), 174-175  
dead.letter file, 793  
Debugging  
  Automounter, 630  
  NFS, 554, 556, 560  
  NIS, 596  
  sendmail, 813  
  UUCP communications, 990  
Default  
  accounts, 67, 81  
  filesystems check, 36  
  home directory, 36  
  mounting using /etc/default/filesys, 551  
  printer, 39, 133, 139  
  remote printer, 513  
  routing, 484  
  security, 81, 83  
  sysadmsh(ADM) options, 40  
  UUCP, 680  
  values, 651  
DEBOOTSTR boot option, 55  
DEL command, DOS, 252  
Delete key, 12  
deliver(ADM)  
  background program, 767  
  checkqueue(ADM), 770  
  fixing mail problems, 958  
delivermail program, 790  
Delivery  
  channels, 757  
  sendmail, 793, 816  
Denylist for printer forms, 132  
Description line (sysadmsh), 19  
Desktop client performance, 880  
Destination, printer, 120  
  /dev/hd0a, UNIX device name, 181  
  /dev/hd0root, running fsck(ADM), 948  
Device  
  cannot access, 987  
  dialout modem problems, 960  
  DOS configuration, 229, 230, 231, 232  
  DOS Services, 217, 218

Device (*continued*)  
  field, Systems file, 672  
  filesystem, 96  
  fixing file permissions, 987  
  locked, 992  
  modem errors, 826  
  naming, 38  
  none available, 960, 988, 992  
  parameters, 925  
  permissions in UUCP, 993  
  printer, 120, 165  
  problem in UUCP, 992  
  specifications, 229  
  sysadmsh(ADM) options, 38  
  UUCP requirements, 663  
Device access  
  canceling specifications, 291  
  "CANNOT ACCESS DEVICE", 987  
  options available, 289  
  preventing, 290  
DEVICE command, DOS, 310, 315, 316, 318  
Device specification file. *See* dosdev file  
Devices file  
  PPP entry, 458  
  baud rate, 666  
  dial-out entries, 683, 835  
  error, 706  
  example control file, 664  
  for TCP/IP, 691  
  format, 683  
  local area network, 673, 684  
  matching Systems file, 663-664  
  modem, 664, 831  
  speed field, 666  
  testing modem, 669  
  UUCP configuration, 662  
  write permissions, 831  
/dev/inet/tcp, 693  
/dev/lp, device, 165  
/dev/nfsd, 557, 559  
/dev/syscon, 965  
df(C), 101, 945  
Dialcodes  
  file, 673, 688  
  UUCP, 670  
Dialer programs, 835  
  binaries, 836  
  /usr/lib/uucp, 664

Dialers file, 688  
   PPP entry, 458  
   Automatic Call Unit (ACU), 684  
   dialer types, 835  
   dialer-token field, 685-687  
   error, 709  
   service code example, 694  
   uucico(C), 664  
 Dialer-token, 683, 686-687, 708  
 Dial-in  
   modem, 830, 834  
   site, 666, 667, 670  
 Dialing out, syntax, 833  
 Dial-out modem, 835  
 Dialup SLIP connection  
   configuring, 444  
   troubleshooting, 445  
 dialups file, 834  
 DIR command, DOS, 291  
 Direct device attachment, 228, 229, 230, 231, 232, 233  
 Direct keyword, 683-684  
 Direct line, 831  
 Directory  
   finding, 103-104  
   GID bit, filesystem, 99  
   problems, cannot remove files, 955  
   size, 102, 109-110  
   sysadmsh(ADM) options, 41  
 disable(C)  
   disabling COM port, 212  
   modem use, 829  
   printer, 134, 137, 215  
 Disabled  
   account, 976  
   terminal, 349, 979  
 Disabling  
   audit, 36  
   C2 features, 927  
   COM port, 212  
   control port, 444  
   printer, 215  
 Disk  
   *See also* Floppy disk; Hard disk  
   adding second, 111  
   bad track table, 942  
   changing layout, 111  
   configuring, 111  
   DOS partition, 219, 220, 221, 222, 224  
   DOS Services, 217

Disk (*continued*)  
   fragmentation, 101, 109, 947  
   free space, 100-101  
   key, 249  
   parameters, 908  
   performance, improving, 872  
   repairing, 111  
   sysadmsh(ADM) selection, 35  
   troubleshooting, 942-944, 947  
   usage, 102  
 Disk drives, DOS, 217, 218  
   setting initial current drive, 294  
 diskcp(C), 174-175  
 Diskette  
   *See also* Floppy disk  
   booting from, under DOS Services, 296  
 dispc, 371  
   terminfo descriptor, 358, 359  
 Dispersion, 521  
 Display  
   console, 13  
   routing table, 437  
   sysadmsh(ADM), 20  
 Display adapters, 341, 342  
   default virtual display for different terminal types, 286  
   DOS Services, 207, 208, 209  
   NLS restrictions, 315  
   NLS sample setup, 316  
   specifying with dos +a option, 286  
   supported under DOS Services, 285  
 display modes, terminfo items, 357  
 Displaying  
   DOS environment variable settings, 335  
   DOS options, with dosopt command, 274, 301  
   dosenv.def file, 274  
   server name, 642  
 Display-oriented DOS programs, 291-292  
 Distributed user account, NIS, 569, 588, 589, 590, 591  
 divvy(ADM), error during installation, 952  
 DIX (thick) cable, configuring, 412  
 "dk\_name - Diskinfo table overflow", 872, 908  
 DMA (Direct Memory Access) channel  
   allowing simultaneous requests, 926  
   tape drive conflict, 980-981  
   transferring buffers requests, 909  
 DMAABLEBUF parameter, 909  
 DMAEXCL parameter, 926

dmesg(ADM), 860

DO387CR3 parameter, 926

## Domain

- file, 750, 753

- fully qualified, 722

- levels, 721-722

- management, 479

- name, 471

- name configuration, 412

- name, registering, 723

- name server, 473

- NIS, 567, 571, 573

- pointer record, 471

- setting up, 464

domainname(NC), 567, 573, 598, 603

## DOS

*See also* DOS Services

- batch files, specifying with dosadmin command, 266

- characteristics, configuring, 263, 264, 265

- coexistence with UNIX, 179

- command interpreter, 293

- configuration file. *See* CONFIG.SYS file

- configuring applications, 207, 245, 247, 248

- copy protection, 248, 249

- defining device names, 187

- device file. *See* CONFIG.SYS file

- display, 207, 208, 209, 210

- Executable File, field, 246

- exit code, setting, 258

- file and directory arguments, 186

- files, 179, 187, 191

- filesystems, 179, 187-189

- floppy disks, mounting, 189

- hard disk devices, 183

- images, 233, 234, 236, 237

- installing UNIX system partition, 184

- interrupts, 256, 332

- links to UNIX system, 252

- partition, 221, 224, 225, 226

  - removing, 185

- primary terminal requirement, 351

- printer output, specifying printer, 215

- program execution, from A: or E: drive, 293

- removing applications, 252

- syntax for specifying options, 262, 263

- sysadmsh(ADM) options, 40-41

- utilities, 179, 186-187

- without DOS Services, 251

- X client, setting, 302

## DOS application programs

- display adapters, compatibility, 342

- installing, 239, 240, 241, 242, 246, 247, 248, 249, 250

- PC scancode terminal, 344

- Protected Mode, 244, 334

- running as the default shell, 339

- running in background, 340

- scheduling, limitations of, 340

- specifying display-oriented, 292

- specifying stream-oriented, 292

- tailoring, 268, 269

DOS character sets, 307, 310

DOS code page, 306, 308, 309, 310, 312, 315, 316, 317, 318, 319

- hardware, 309

- software, 309

- switching, 310

dos command, DOS Services

- +a option, 208

- listing available options, 295

- specifying DOS images, 296

- syntax, 262, 281, 282, 283

## DOS commands

- ATTRIB, 222

- BACKUP, 217

- BUFFERS, 235

- CHCP, 306, 310, 314, 315, 316, 321

- CHKDSK, 332

- compatibility with DOS Services, 332

- COPY, 299

- COUNTRY, 306, 310, 314, 315, 316

- DEL, 252

- DEVICE, 310, 315, 316, 318

- DIR, 291

- DOS, 235

- FCBS, 235

- FDISK, 332

- FORMAT, 251, 332

- KEYB, 306, 310, 314, 315, 316

- MODE, 217, 306, 310, 314, 315, 316, 318, 321

- NLSFUNC, 306, 310, 316, 318

- PRINT, 299

- RESTORE, 217

- SORT, 291

- STACKS, 235

- SUBST, 241, 242

- SWITCHES, 235

- SYS, 332

- TREE, 291

DOS commands (*continued*)

TYPE, 292

## DOS images

defaults, 296

DOS Services configuration for, 296

specifying with dos command, 296

specifying with dosopt command, 296

starting environment without, 296

## DOS memory

changing allocations for startup, 269

specifying amount, 297

## DOS options

*See also* dosopt command

assigning with dosopt command, 271, 272

changing with dosadmin command, 265

creating a null record, 302

displaying values, 301

help screen, 295

removing assigned values, 302, 303

running autoexec.bat, 298

specifying on DOS command line under DOS Services, 262, 263

## DOS partition

defined, 219

extended, 219

installing applications on, 249, 250

physical, 219, 220, 221, 222, 285, 288, 290

virtual, 220, 223, 224

## DOS Services

*See also* DOS

application programs, 245, 247, 248

backup, 217

disk drives, 217

display, 208, 209, 210

environment variable, XMERGE, 330

files affected by common administrative procedures, 238

installing applications, 239

problems, 333

## DOS startup file

*See also* AUTOEXEC.BAT and CONFIG.SYS files

specifying interpretation of, 269

specifying with dos command, 294, 298

specifying with dosadmin menus, 270

dos2unix command, DOS Services, 306, 322

dosadmin command, DOS Services, 263

changing DOS options, 265

configuring applications, 239, 240, 245, 246, 247, 249

dosadmin command, DOS Services, 263

*(continued)*

defined, 206

differences from dosopt command, 271

DOS images, 236, 237

menu, 246, 247, 269

specifying DOS startup file, 266

specifying startup memory, 265

tailoring DOS application programs with, 269

tailoring DOS environment with, 269, 271

tailoring DOS operation with, 265, 267, 268

virtual diskette, 226, 227

virtual partition, 224

dosboot command, DOS Services, 297

booting from a virtual diskette, 228

dosdev file, 289

device specification file, 231, 232

finding the device name for the extended partition, 221

setting up, 231

syntax, 231, 232

DOSENV environment variable

how to set, 337, 338

using in UNIX .profile file, 338

dosenv.def file

creating personal, 273

differences between personal and system, 264

displaying personal, 274

displaying system, 274

purpose, 266

removing installed options, 275

restoring installed option files, 275

uses for personal, 264, 272

uses for system, 272

DOSINODE parameter, 915

"dosiread - DOS inode table overflow", 872

doslp, printer, 213

DOSMOUNT parameter, 915

dosopt command, DOS Services, 232, 263, 271, 272, 273, 274, 281, 282

automatic attachment of DOS partition, 225

configuring DOS applications, 248

differences from dosadmin command, 271

displaying assignments of DOS options, 301

displaying current DOS options, 274

DOS processes controlled by, 272

example uses of, 273, 274

dosopt command, DOS Services, 232, 263, 271, 272, 273, 274, 281, 282 (*continued*)  
 explained, 272  
 installing and removing DOS options, 271, 272, 273, 274  
 listing available options, 295  
 null option, 302  
 removing assigned values, 302, 303  
 restoring DOS application default values, 275  
 specifying DOS images, 296  
 syntax, 272, 274, 281, 282, 283  
 DOSPATH environment variable  
   how to set, 337  
   when to use, 336  
 d\_passwd file, 834  
 DQP-10 matrix printer, 122  
 Drift, 521  
 driftfile, 533  
 Driver  
   configuring, 411  
   kernel, 423  
 DSTFLAG parameter, 918  
 du(C), 102, 946  
 dumb, printer, 122

## E

E: drive, 250  
 EBUSY error, 913  
 Echo problems  
   modem, 960, 962  
   terminal, fixing, 985  
 Editing  
   command line, 12, 13  
   sysadmsh(ADM) option, 41  
 EDU domain name, 722  
 Effective buffer size, 650  
 EGA card  
   characteristics and options, 342, 343  
   requirements, 342  
   UNIX MultiScreen functionality, 344  
 80387 math coprocessor, 926  
 eisa(ADM), 54  
 elite pitch, 133, 158  
 Emergency Boot Floppy Set, 932  
   restoring root filesystem, 948  
   restoring system files, 932-934, 936-938

EMS  
   configuration for, 331  
   DOS Services command option for, 287  
 emulator, printer type, 122  
 enable(C)  
   COM port, 212  
   modem port, 833  
   printer, 134, 137  
 Enabling  
   audit, 36  
   COM port, 212  
   disabled account, 976  
   export, 549  
   printer, 216  
   terminal, 979  
 Enter key, 12  
 Enter new time ([yymmdd]hhmm): message, 46  
 Environment variable, DOS  
   CONVOPTS, 323, 324  
   displaying settings, 335  
   how to set, 336  
   PROMPT, 338  
   setting from UNIX shell, 337  
   setting in UNIX .profile file, 338  
 Environment variable, DOS Services,  
   XMERGE, 330  
 Environment variable, UNIX  
   CODEPAGE, 314, 317, 319, 322  
   CONVOPTS, 324  
   COUNTRY, 314, 317, 319  
   DOENV, 337, 338  
   DOSPATH, 336, 337  
   ISO\_TERM, 319  
   KEYB, 314, 317, 319  
   LANG, 310, 311, 314, 319, 322  
   passing to DOS, 335  
   PATH, 868  
   setting with automount(NADM), 624  
   TZ (timezone), 918  
 Epson printer, 122  
 Equivalence, 417, 429  
 Erase key, 13  
 Error, mailer, 804  
 Error checking (fsdb), disabling, 954  
 Error Correction Code (ECC), 172  
 Error correction, modems, 963  
 Errorlogs, 860



## Error messages

Automounter, 633

NFS, 557, 558, 559, 560

NIS, 598, 599

## Errors-To header line, 822

## Escape, login script, 674

## Escape key, 12

## Escape sequences

defined, 354

special action, 364

/etc directory, network files, 429

/etc/auth/system/default file, 59

/etc/auth/system/gr\_id\_map file, 50

/etc/auth/system/pw\_id\_map file, 50

/etc/auth/system/ttys file, 978

/etc/auto.master. *See* Maps, Automounter

/etc/bcheckrc file, 937

/etc/cleanup script, 108

/etc/conf/cf.d, 646, 933

/etc/conf/init.d/sio, 830

"/etc/conf/pack.d/ct/space.c line 46 bad octal digit", 980

/etc/cshrc, in an DOS Services NLS environment, 314

/etc/ddate file, 106-107

/etc/default directory, 54, 169, 186

/etc/default/authsh file, 67

/etc/default/cleantmp file, 100, 946

/etc/default/filesys, 98, 547, 551, 553, 556, 625, 634

/etc/default/format file, 174

/etc/default/goodpw file, 89

/etc/default/lang, 319

/etc/default/login file, 58

/etc/default/lpd, 164

/etc/default/msdos file, 187

/etc/default/passwd file, 58, 89

/etc/default/tar file, 169

/etc/dosdev file. *See* dosdev file

/etc/d\_passwd, 834

/etc/emulator file, 934

/etc/exports, 547, 549

/etc/ftpusers, 417, 430

/etc/gated.bgp, 485

/etc/gated.conf, 485

/etc/gated.egp, 485

/etc/gated.hello, 485

/etc/gated.rip, 485

/etc/gettydefs file, 964

/etc/group file, 76-77, 570, 697, 866, 956

/etc/hosts, 416, 480, 547, 550, 558, 559, 573, 614, 631, 636

/etc/hosts.equiv, 417, 429, 511, 514

/etc/hosts.lpd, 511, 514

/etc/inetd, 428

/etc/inetd.conf, 428, 690

/etc/init file, 938

/etc/inittab file

editing, 830

recreating, 936

setting baud rate in, 964

UUCP, 698, 700

/etc/mnttab, 547, 557, 627

/etc/motd file, 36, 100

/etc/named.pid, 480

/etc/networks, 446

/etc/nfs, 549, 581

/etc/passwd file, 58, 60, 80, 419, 569, 570, 639, 697, 699, 834, 866

/etc/passwd.local, 569, 591

/etc/passwd.yp, 569, 591

/etc/printcap, 511

/etc/profile

in an DOS Services environment, 317

in an DOS Services NLS environment, 314

/etc/rc\* scripts, 45

/etc/rc2.d scripts

S86mmdf, 767

S87USRDAEMON, 694

starting cron daemon, 941

starting deliver daemon, 958

starting lpsched daemon, 967

/etc/resolv.conf, 474

/etc/services, 691, 692

/etc/shadow file, 58, 60

/etc/shutdown program, 115

/etc/systemid file, UUCP, 698

/etc/wtmp file, 107-108

/etc/yp, 566, 574

/etc/yp/Makefile, 582, 583

/etc/ypservers, 573

/etc/yp/ypmake, 582, 594

/etc/yp/Y\_P\_MAP\_X\_LATE, 570, 573, 587

Ethernet/SLIP gateway, 446

ETRUNC parameter, 99, 913

EVDEVS parameter, 925

EVDEVSPERQ parameter, 926

Event, audit, 37

"event - Event channel full", 872, 926

"event - Event table full", 872, 925

EVQUEUES parameter, 925  
 Exabyte (8mm) tape drive, size, 170  
 Exclusive device access, specifying, 289  
 execsuid authorization, 92  
 Executing  
   recursive commands, 103-105  
   remote programs, 665  
   system programs, 37  
 Exit code  
   DOS, setting, 258  
   printer, 160  
 Exiting  
   shell, 13  
   sysadmsh(ADM), 42  
 Expanded memory. *See* EMS  
 Expanding, system, 14  
 Expect string, 673-674  
 exportfs(NADM), 550  
 Exporting, filesystem, 547, 549  
 exports(NF), 555  
 EXTRA\_NDEV parameter, 929  
 EXTRA\_NEVENT parameter, 929  
 EXTRA\_NFILSYS parameter, 929

## F

F1 function key, simulating on ASCII  
 keyboard, 347  
 Failure mode, 83  
 "falloc - File table overflow", 872, 913  
 Fault  
   alerting, 130  
   printer, 129-131  
 FCBS command, DOS, 235  
 FDISK command, DOS, 332  
 fdisk(ADM)  
   main menu, 180  
   partition, 181, 185  
   switching operating systems, 183  
   UNIX system and DOS versions, 179  
 fhandle, 555, 557  
 File  
   *See also* Filename; Filesystem  
   access, 417  
   access operations, sar -a, 891  
   archiving, 167, 170, 175  
   audit system, 107  
   cannot remove from directory, 955  
   checking MMDF configuration files, 958

File (*continued*)  
   constable, 646  
   copying remote, 664-665  
   corruption, 48-50  
   DOS, 41  
   DOS Services, 238  
   /etc/conf/cf.d, 646  
   finding, 103-106  
   GID bit, 99  
   handle, 555, 557  
   last access time, 103  
   lmxrc, 641, 646  
   locating, 946  
   locking, 562  
   maximum number of open, 913  
   modes, sendmail, 816  
   monitoring, 107  
   mtune, 646  
   name conversion, 652  
   NFS, 562  
   parameters, 912  
   permissions, 222  
   process accounting, 107  
   recursive removal, 105, 106  
   repairing, 116  
   restoring, 51, 932  
   sharing parameters, 651  
   size, 102  
   size limit, 916  
   structures, 112  
   stune, 646  
   sysadmsh(ADM) options, 41  
   transfer, UUCP, 659, 662-665, 702  
   transferring ownership, 104  
   /usr/lib/lm, 646  
   UUCP log files, 107  
 File table  
   allocating open entries, 913  
   status, 898  
 Filename  
   mapping in a multilanguage environment,  
     308, 320  
   truncation, 99  
 "Filename too long", 913  
 filesys(F), 555  
 Filesystem, 95  
   adding, 96, 111  
   avoiding problems, 859  
   checking, 36, 114, 117  
   cleaning, 44

Filesystem, 95 (*continued*)  
     configuring, 99  
     corruption, 47-50, 111, 112, 114  
     creating, 109  
     default mounting, 551  
     expanding, 111  
     exporting, 549  
     floppy disk, 176-178  
     free space, 100-101, 977  
     maintaining, 108  
     mounting, 97, 556, 557, 945  
     parameters, 912  
     remote, 562, 563  
     removing, 109, 114  
     repairing, 115  
     root, 95, 115  
     security, 886  
     structures, 112, 113  
     sysadmsh(ADM) options, 36, 41, 96-97  
     troubleshooting, 934, 944-955  
     unmount, 945  
     user mounting, 553  
     writing buffers to disk, 910  
 Filter, printer, 148, 152  
 find(C), 103-106, 946  
 Finding files, 103-104  
 FIPS 151-1 conformance, NIST, 92  
 Fixed disk, 252  
 fixmog(ADM), 62, 866  
 fixperm command, checking UUCP file permissions, 994  
 FLCKREC parameter, 914  
 Floating-point emulator error, 934  
 Floating-Point Unit (FPU), 54  
 Floppy, sysadmsh(ADM) selection, 178  
 Floppy disk, 173, 175  
     *See also* Disk, 173  
     booting, 933  
     copying, 174-175  
     drive, DOS Services, 217  
     filesystem, 176-178  
     formatting, 173  
     mounting, 217  
     sysadmsh(ADM) options, 40, 176  
     virtual, 223, 226, 227  
 Fonts, 125-128, 330  
 Forcing mail queue, 813, 819  
 Forking during queue runs in sendmail, 815  
 Form printer, 140, 148

Format  
     mail, 713  
     mailbox, 772  
 FORMAT command, DOS, 251, 332  
 Formatting  
     floppy disks, 173  
     floppy disks, sysadmsh selection, 40  
     forward files, 822  
 Forwarding, 794  
 Fragmentation, 101, 109  
 Free list, 117  
 Free space, 100-101, 111  
 Free-block list, 117  
 fsck, fileys(F) entry, 552  
 fsck(ADM), 111, 115-116  
     boot option, 44  
     checking hard disk problems, 948, 951  
     DOS, use with, 189  
     failed mount command, 954  
     phases, 116  
     restoring root filesystem, 948  
     single-user mode, 865  
     system crash, 886  
     troubleshooting considerations, 859  
     wrong ISIZE and FSIZE values, 952  
 fsdb(ADM)  
     correcting super block values, 953  
     disabling error checking, 954  
     DOS filesystems, 954  
     filesystem debugger, 952  
     repairing DOS filesystem, 116  
     repairing UNIX filesystem, 116, 954  
 FSIZE value, 952-953  
 "fsstat: root filesystem needs checking", 44  
 fstyp, fileys(F) entry, 552  
 ftp account, 417, 430

## G

Games  
     booting from diskette, 296  
     game port, 287  
     running under DOS Services, 297  
 "Garbage or loose cable ... port shut down", 963  
 gated(ADMN) daemon, 483  
 Gateway  
     defined, 483  
     MMDF, 753

Gateway (*continued*)

- smart, 484
- gethostbyname call, 480
- getid (ADMN), 499
- getmany (ADMN), 500, 501
- getone (ADMN), using, 499
- getty(M), process, 212
- goodpw(ADM), 89
- GOV domain name, 722
- GPGSHI parameter, 917
- GPGSLO parameter, 917
- GPGSMK parameter, 917
- Graphical Environment, streams, 922
- Graphics parameters, sysadmsh(ADM)
  - selection, 36
- gr\_id\_map file, 50
- Group, 76-77
- Group ID (GID)
  - filesystem, 99
  - network requirements, 419
  - setting directory SGID bit, 99
  - supplemental groups, 77
- Guest
  - ftp account, 417
  - UUCP transaction, 665

## H

Hard disk

- See also* Disk
- adding, 36
- DOS partition, 219-224
- DOS Services, 217
- partitions, 180-182, 184, 185, 220
- tracks, 181
- troubleshooting, 935, 948

Hard mount, 551, 555

Hardware devices

- accessible to DOS, 284
- accessing DOS partition, 288
- attaching COM ports to DOS, 286
- attaching printer ports to DOS, 287
- default display types, 286
- displaying, 53-54
- errors, 860
- game port, 287
- keyboard, 284
- mouse, 285
- preventing access, 290

Hardware failure

- defective printer, 968, 970
- modem port problems, 960
- non-functional terminal, 982
- super block damage, 951
- system panic, 864
- tape controller card, 980-982
- terminal, hung, 983-984

Hardware-dependent parameters, 926

Hash queues

- increasing with system buffers, 908
- S5HASHQS parameter, 915

Hayes (and compatible) modem

- configuring, 828
- connecting, 666
- switch settings, 826-827
- testing, 668

hd(C), 110

Header lines, 822

Help screen, for dos and dosopt commands, 295

Hercules adapter, option for specifying, 286

Hierarchy, sysadmsh(ADM), 35

Highlighting, sysadmsh, 20

/home, 614, 616

Home directory

- changing, 70
- sharing, 66
- sysadmsh(ADM) selection, 36
- user, 57

HOME key, simulation on ASCII keyboard, 347

homepaths file, 78

Host

- adapter number, SCSI tape drive, 981
- defined, 408, 521
- information resource record, 470
- UUCP transaction, 665

Host name

- changing, 761
- hiding, 724
- setting, 411
- underscore character, 721

hostname(TC), 479

Hosts

- database, 416
- file, 636

hosts.byname map, 614, 631

hosts.equiv file, 417, 429

hosts(NF), 550

HP printers, 122  
 "hsiread - extent table overflow", 872  
 "hsiread - inode table overflow", 872  
 HSNEXTENT parameter, 915  
 HSNINODE parameter, 915  
 HSNMOUNT parameter, 915  
 hwconfig(C), 54  
 Hyphen character (-), 300

## I

IBM Proprinter XL, 122  
 ICMP, routing-redirect messages, 484  
 "id - out of STREAMS queues", 872  
 "id - out of STREAMS queues", 923  
 ID number  
   SCSI tape drive, 981  
   virtual circuit, 642  
 Identity, sysadmsh(ADM) selection, 69-70  
 Idle time, sysadmsh(ADM) selection, 36  
 idmkenv(ADM), 936, 973  
 idtune(ADM)  
   kernel parameters, 887, 930  
   modifying parameters with, 652, 658  
 ifconfig(ADMN), 427, 449  
 "ifreeget - Inode table overflow", 872, 912  
 ignore flag, 534  
 Ill-behaved DOS programs. *See* display-oriented DOS programs  
 Images, DOS, 233, 234, 236, 237  
 Immediate device access, specifying, 289  
 inetd(SFF), 428, 690  
 "INIT: execlp of /bin/sulogin failed", 937  
 Initializing  
   cache, 466  
   NIS, 574  
 init(M)  
   boot problems, 936, 938  
   maintaining filesystems, 114  
   S command, 965  
 "INITSH: /etc/bcheckrc not found", 937  
 inittab file, 830  
 Inode table  
   allocating entries, 912  
   status, 898  
 Inodes, 113  
   *See also* Inode table  
   allocation, 952  
   available, 953

Inodes, 113 (*continued*)  
   clearing, 116  
   count error, 112  
   directory, 109  
   filesystem problems, 944  
   null numbers, 110  
   parameters, 912  
   reconfiguring with mkfs(ADM), 945  
   shadow, 110  
   structure, 113, 658

## Installing

  copy-protected applications, 248, 249  
   DOS applications, 239, 240, 242, 244, 246, 247, 248, 249, 250, 251, 334  
   Emergency Boot Floppy Set, 932  
   fonts, 125-128  
   modem, 825-826, 828  
   planning system, 14  
   PPP, 450  
   printer, 123-128, 164  
   remote printing, 508  
   SLIP, 441

  integrity(ADM), 50, 62, 596, 887

## Interface

  network, 427, 436  
   printer, 121, 156-162

Internal modem, 829

International Organization for Standards. *See* OSI

## Internet, 711

*See also* DARPA

  address, 427  
   address configuration, 413, 414  
   daemon, 428  
   defined, 408, 521  
   protocol, 408  
   technical bulletin RFC822, 758  
   top-level domain codes, 722

InterNIC, 483, 723

InterNIC Information Services. *See* InterNIC

Interoperability, NIS, 601

Interprocess communication, sar -m, 894

Interrupt key, 12

Interrupt rate, clock, setting with merge  
   fastclk option, 254

## Interrupt vector

  boot display, 54  
   COM ports, 832  
   configuring, 411  
   conflicts, 828

Interrupt vector (*continued*)

- printer port conflict, 968
- tape drive conflict, 980, 982

## I/O

- configuring base address, 411
- tuning, 872

## IP

- address, 637
- defined, 408

IRQ (physical interrupt). *See* Interrupt vector

## Irwin, tape drive, 172

## ISIZE value, 952-953

## ISO

- characters, 360, 362
- Open Systems Interconnection(OSI), 404

## ISO terminal

- DOS Services NLS setup procedures, 319
- in an DOS Services NLS environment, 306, 313, 319
- using with DOS, 360, 362

## ISO\_TERM environment variable, UNIX, 319

## J

Job. *See* Process

## Jumper settings, tape drives, 980, 981

## K

## KDBSYMSIZE parameter, 912

## Kernel

- authorizations, 90-93, 942
- configuration, 423
- environment, 936
- managing virtual address space, 916
- mounting filesystems, 555
- parameters. *See* Kernel parameters
- relinking with link\_unix, 882
- resources, 882, 887
- sysadmsh(ADM) selections, 35

## "Kernel: i/o bufs", 938

## Kernel parameters, 905, 907

*See also* Parameters; System

- asynchronous I/O, 928
- boot load extension, 929
- buffers, 908
- changing, 882, 905
- clock, 918
- devices, 925

Kernel parameters, 905, 907 (*continued*)

- disks, 908
- ETRUNC, 99
- event queues, 925
- files, 912
- filesystems, 912
- idtune(ADM), 887
- inodes, 912
- LAN Manager, 930
- memory management, 915
- message queues, 919
- modifying, 580, 652, 655
- multiprocessing, 929
- multiscreens, 918
- overriding limits, 883
- processes, 915
- semaphores, 920
- shared data, 921
- streams data, 922
- swapping, 915
- sysadmsh(ADM), 35
- system name, 922
- table overflow, 871-872
- TCP/IP, 930

## Kernel problems

- backing up, 940
- booting from alternate kernel files, 933
- inconsistencies causing system panic, 864
- keyboard lockup patch, 940
- out of inodes on EAFS /dev, 944

## Key, ID, 542

## Key disk, 249

## Key sequences

- emulating missing keys, 355, 363, 365
- escape, 354
- special action, 364

## Key simulation on ASCII keyboard, 348

## KEYB command, DOS, 306, 310, 314, 315, 316

## KEYB environment variable, UNIX, 314, 317, 319

## Keyboard

- code, 316
- DOS, 333
- end input, 13
- lockup, 939-941, 974, 975
- mapping on a serial terminal, 345, 347
- polling, 254
- test with modem, 832
- type, console, 941
- under DOS Services, 284

Keyboard (*continued*)  
 UNIX keystrokes, 12  
 KEYBOARD.SYS file, 318  
 Keystrokes, 12  
 Kill key, 12, 13  
 Killing, DOS processes, 349

## L

LAN (Local Area Network)  
*See also* Local  
 defined, 402  
 MMDf, 752  
 planning, 15  
 UUCP, 673, 683-684  
 LAN Manager Client, 419  
 Landscape mode, 122  
 LANG environment variable, UNIX, 310, 311, 314, 319, 322  
 LCK (lock) files, 664, 669, 702-703, 705, 709  
 Lead-in key sequence  
 changing, 366, 373  
 default, 373  
 defined, 354  
 Leased line, 667  
 Lineprinter, locating, 15  
 Link, DOS/UNIX, 252  
 link\_unix(ADM), 882  
 list channel, 748, 758  
 List owners, 821  
 listen, UUCP, 694  
 Listing  
 directory, sysadmsh selection, 41  
 dos command options, 295  
 dosopt command options, 295  
 lmxrc file  
 modifying, 641  
 network buffer size, 650  
 operating parameters, 641, 646  
 parameter description, 652  
 purpose, 651  
 ln(C) command, configuring DOS applications, 248  
 Load image. *See* DOS images  
 Load management, printer, 141  
 Loading, operating system, 44  
 Local  
*See also* LAN  
 channel, 758

Local (*continued*)  
 domain, 722  
 drives, interpreting as remote, 255, 256  
 files, interpreting as remote, 256  
 name table, 643  
 network switch, 686-687  
 subnetwork, 427  
 systems, connecting with direct wire, 695  
 user account, NIS, 569, 588, 590, 591  
 Locale variables, sysadmsh(ADM) selection, 36  
 Locating files, 103-104, 105  
 Lock files, UUCP, 993  
 Locked  
*See also* Keyboard  
 account, 956  
 device, 992  
 keyboard, 939-941  
 terminal, 985  
 Locking  
 account, 68, 976  
 terminal, 57  
 Log book, 10  
 .Log directory, UUCP programs, 991  
 Log files  
 cleaning, 106-108  
 clearing system files, 946  
 /etc/ddate file, 106  
 monitoring, 107  
 printer, 149-150  
 /usr/adm/sulog file, 106  
 UUCP, 661, 662, 702, 991  
 logger(TC), 860  
 log(HW), 860  
 Login  
 configuring, 65-66, 85-86  
 creating dial-in accounts, 667  
 directory size, 110  
 ending session, 13  
 multiscreen, 14  
 prompt, 52  
 reports, 38  
 script (chat script), 670, 673-674  
 security, 38, 81, 83, 85-86, 689  
 sequence, UUCP, 662, 664, 674  
 shell, adding, 94  
 system boot, 45  
 troubleshooting, 935, 955-957, 982, 988  
 User ID (LUID), 927  
 "Login incorrect", 978

*login:*

"login: resource Authorization name file could not be allocated ...", 978  
login(M), restoring, 935  
Logins, sysadmsh(ADM) selection, 68, 85-86  
LOGNAME entry, Permissions file, 677  
Loop, process in, 967  
lost+found directory, 97, 117  
low on async cells (NASEVENT) message, 655  
low on bindxs (NBINDX) message, 656  
lowpriotrap flag, 534  
lp authorization, 91  
lp\* commands, 135  
lp(C), 510  
lpd(ADMN), 509  
LPDEST, shell variable, 139  
lpmove(ADM), 143  
lpq command, 511  
lpr command, 511  
lprm command, 511  
lpsched(ADM)  
    enabling and disabling printers, 137  
    manually starting the print service, 136  
    print problems, 967  
    starting, 967, 969  
lpshut(ADM), daemon, 136  
lpstat(C), 133, 147-148  
LPTELL shell variable, 161  
LQP-40 printer, 122  
LUID (Login User ID), 78  
LUN (Logical Unit Number), SCSI, 981

## M

MACHINE entry  
    OTHER option, 682  
    Permissions file, 677  
Machine name, *See also* Host name  
Macro, defined, 797  
Mail, 711  
    *See also* MMDF  
    administering, 770  
    header, 713, 794  
    queue, 818, 819  
    resource record, 472, 473  
    routing, 714, 750  
    sending to remote sites, 659  
    troubleshooting, 957-959  
    UUCP, 662, 670

Mail Transport Agent (MTA), 711  
Mail User Agent (MUA), 711  
Mailbox  
    file, missing, 958  
    resource record, 472  
Mailer  
    defining, 798  
    flags, 803, 811  
Maintenance, UNIX, 9, 10  
make(CP), 582, 583, 594  
makedbm(NADM), 582, 585, 586, 587, 592, 593, 594, 603  
MakeDir DOS, sysadmsh(ADM) selection, 41  
Management Information Base (MIB), 494  
Management stations, 495  
Managing  
    client, 642  
    user accounts, 548  
map2ascii(NADM), 585, 601, 602, 603  
mapflush daemon, 916  
Mapping, keyboard on a serial terminal, 345, 347, 348  
Maps, Automounter  
    direct, 611, 618, 628  
    indirect, 611, 616, 627  
    master, 611, 613, 627  
Maps, NIS  
    Automounter, 613  
    creating, 585  
    defined, 566  
    different versions, 596  
    hosts.byname, 614, 631  
    managing, 582  
    modifying, 583  
    propagating, 567, 584, 587  
Master server  
    NIS, 567  
    primary, 462, 474  
    secondary, 462, 474  
MAXACPU parameter, 929  
MAX\_BDEV parameter, 929  
MAXBUF parameter, 909  
MAX\_CDEV parameter, 929  
MAX\_CFGSIZE parameter, 911  
MAX\_CONFS\_SESS, 647, 652  
MAXFC parameter, 917  
MAXPMEM parameter, 916  
MAXSC parameter, 917  
MAXSEPGCNT parameter, 924  
MAXSESSIONS, 646, 655



maxskew, 540  
 MAXSLICE parameter, 918  
 MAXUMEM parameter, 917  
 MAXUP parameter, 916  
 MCHANLOG, 785  
 MDA, option for specifying, 286  
 MDLVRDIR, 772  
 Media  
   administering, 15  
   sysadmsh(ADM) options, 40  
   types, 167  
 mem authorization, 91  
 Memory  
   image, 860  
   management, 915  
   shared segment size, 921  
 Memory, startup  
   acceptable values for, 270  
   specifying with dosadmin command, 265  
 Menu, sysadmsh(ADM), 19-20, 35  
 menumerge(ADM), 34  
 merge command  
   display option, 258  
   drive option, 255, 256  
   fastclk option, 254  
   handle option, 256  
   pollsleep option, 254  
   return option, 258  
   syntax, 253  
   uses for, 253  
   using batch files with, 258, 259  
 mesg(C), 128  
 Message  
   files, 311  
   Processing Module (MPM), 791  
   queues, 795, 919  
   timeouts in sendmail, 815  
   UUCP, 988  
 messages file, 860  
 Metacharacters, Automounter, 622  
 MFAILTIME, 784  
 MIL domain name, 722  
 MINARMEM parameter, 917  
 MINASMEM parameter, 917  
 MINHIDUSTK parameter, 917  
 Mini-cartridge, sizes, 170  
 Minus symbol (-), in DOS Services commands, 281  
 MINUSTKGAP parameter, 917  
 mkcon, 640

mkdev(ADM)  
   aio, 928  
   fs, 96-97  
   nis, 573, 574  
   ptty, 919  
   rlp, 508  
   snmp, 497, 498  
   tape, 168  
 mkfs(ADM)  
   backups, 945  
   reconfiguring inodes, 945  
 mking command, DOS Services, 318  
   DOS images, 236  
 MLCKTYPE, 784  
 MLDOMAIN, 742  
 MLNAME, 742  
 MLOCMACHINE, 742  
 MMAXHOPS, 785  
 MMBXNAME, 772  
 MMBXPROT, 772  
 MMDF  
   *See also* Mail  
   administering, 770  
   compared to sendmail, 790  
   configuring, 716, 724, 739, 742, 750, 752-753, 772  
   delivery, 716  
   hidden machine, 724  
   host name, 761  
   keywords, 742, 772  
   LAN (Local Area Network), 752  
   mail format, 713  
   mailbox file, 772  
   maintaining, 770  
   parameters, 772, 782, 784-785  
   problems, 957  
   queues, 770  
   routing, 713-714, 750, 759, 772  
   testing addresses, 770  
   testing configuration, 769  
   UUCP, 670  
   UUXSTR, 766  
   XENIX file conversion, 763  
 MMDF alias  
   *See also* Alias file  
   ALIAS entry, 782  
   creating file, 781  
   file, 747  
   search sequence, 750  
   table, 781-782

- MMDF channel, 757
  - badhosts, 716, 758
  - bad users, 716
  - configuring, 753
  - list channel, 748
  - types, 758
- mmndfalias(ADM), 764
- mmndftailor(F), 739, 958
- MMSGLOG, 785
- MNP4, 963
- /mnt directory, 176
- mnt(C), 98, 553
- mnttab(NF), 555
- Mode
  - changing, 12
  - single user, 45-46
  - system maintenance, 45-46
- MODE command, DOS, 217, 306, 310, 314, 315, 316, 318, 321
- Mode NTP packets, 542
- Modem, 825
  - auto-answer, 833
  - baud rate, 666
  - checking baud rate, 960
  - checking /etc/gettydefs file, 964
  - checking modem cable, 959
  - configuring, 826-828
  - connecting, 665, 666
  - data compression, 963
  - dialers, 835
  - Dialers file, 664, 686-688
  - dial-in, 833-834
  - dialing configuration, 827
  - dial-out, 833, 835
  - editing /etc/inittab file, 830
  - error correction, 963
  - Hayes (and compatible), 666, 668, 826-827
  - incompatible, 963
  - installing, 825-826, 828
  - local network switch, 687
  - login sequence, 674
  - null modem, 829
  - pin connections, cabling, 829
  - planning, 15
  - printer, 163
  - printer connection, 125
  - problems in UUCP, 992
  - speed conversion, 839
  - supported, 661
  - switch settings, 827

- Modem, 825 (*continued*)
  - telephone line, 659
  - testing, 668, 831
  - testing phone line, 960, 992
  - Trailblazer, 828
  - troubleshooting, 959-965
  - UUCP use, 829
  - volume, 828
- MODE\_SELECT parameter, 926, 974
- Monitoring NTP, 543
- Monochrome display adapter. *See* MDA
- Motif libraries performance, 880
- mount, filesystem(F) entry, 552
- Mount table, 547, 555, 557, 627, 913
- mount(ADM)
  - boot floppy, 933
  - directory, 547
  - error message, 557
  - failed, 954
  - hard disk root filesystem, 935
  - remote filesystem, 551
  - use of, 97, 178
- mountdir, filesystem(F) entry, 552
- mountd(NADM), 548, 555
- Mounting, 97
  - filesystem, 96-98
  - floppy disk directories, 176-178
  - floppy disk, DOS Services, 217
  - print wheel, 127-128, 139-140
- Mounting, Automounter
  - direct, 611
  - hierarchical mounts, 619, 632, 634
  - included maps, 625
  - indirect, 611, 612
  - mount options, 613, 616, 618, 626, 630
  - mount points, 613, 614, 616, 618, 626, 628, 632, 634
  - multiple mounts, 618
- Mounting, NFS
  - directory, 547
  - hard, 551, 555
  - mount points, 551, 553
  - remote filesystem, 551, 555, 556, 557
  - soft, 551, 555
- mount(NADM), 547, 551, 555, 556, 557, 607, 613
- mount(NS), 555
- Mouse, 285
  - installed incorrectly, 966
  - installing, 36

Mouse, 285 (*continued*)  
     troubleshooting, 965, 966  
     /usr/lib/event/ttys, 966  
 Moving, print jobs, 143  
 MPM (Message Processing Module), 791  
 "MPX timeout - Timeout table overflow", 872, 918  
 MSGMAP parameter, 919  
 MSGMAX parameter, 919  
 MSGMNB parameter, 919  
 MSGMNI parameter, 919  
 MSGSEG parameter, 920  
 MSGSSZ parameter, 919  
 MSGTQL parameter, 919  
 MSLEEP, 785  
 MSUPPORT, 742  
 MTA (Mail Transport Agent), 711  
 mtune(F)  
     parameter, 646, 652, 653, 655, 656, 657, 658  
     parameters file, 646, 907  
     streams buffer, 651  
 MUA (Mail User Agent), 711  
 Multilingual code page, 309  
 Multiple locations, 620  
 Multiplexer links, 923  
 Multiport serial cards, 331, 826  
 Multiprocessing, kernel parameters, 929  
 MultiScreen, 13, 14  
     diagnosing keyboard lockup, 939  
     DOS Services, 209  
     example, 14  
     interaction with EGA card, 344  
     interaction with VGA card, 344  
     parameters, 918  
     troubleshooting, 978  
 Multiuser mode, 45  
     cannot enter, 936  
     no login on console, 941  
     system hangs at login prompt, 935  
 MWARNTIME, 785  
 mwm(X), 880

## N

NAHACCB parameter, 927  
 NAIobuf parameter, 928  
 NAIoHBuf parameter, 928  
 NAIoLockTBL parameter, 929  
 NAIoProc parameter, 928

NAIOREQ parameter, 928  
 NAIOREQPP parameter, 928  
 NALIAS parameter, 655  
 Name cache, 895  
 Name resolution, 536  
 Name resource record, canonical, 471  
 Name server  
     cache initialization, 466  
     caching-only, 463, 474  
     changing origin, 469  
     data files, 467  
     defined, 461  
     master servers, 462  
     multiple files, 468  
     remote, 463, 466  
     resource records, 467-473  
     sample files, 473, 477  
     slave, 463  
     starting, 464, 479  
     types, 462  
 named(ADMN), 479, 480  
 named.hosts, 476  
 named.local, 475  
 named.rev, 477  
 Naming files, sysadmsh(ADM) selection, 41  
 NASEVENT parameter, 655  
 National Language Support, *See* NLS, 305  
 NAUTOUP parameter, 909  
 nbconf file, 638  
 NBIDS parameter, 655  
 NBINDX parameter, 656  
 NBLK parameter, 922  
 NBLKn parameter, 652, 923  
 NB\_SCOPE parameter, 638  
 nbstatus command, 637, 642, 643  
 NBUF parameter, 908  
 NCALL parameter, 918  
 NCALLRETRY parameter, 656  
 NCB name length, 657  
 NCLIST parameter, 911  
 NCONSTABLE parameter, 656  
 NCOPYBUF parameter, 909  
 NCPYRIGHT parameter, 911  
 NDISK parameter, 908  
 NEMAP parameter, 911  
     /net, 614  
 net commands, 640, 641, 642, 650  
 NET domain name, 722

## NetBIOS

- activity, 643
- checking status, 643
- configuring with TCP/IP, 636
- defined, 635
- scope identifier, 638
- sessions, 635, 645, 646, 655
- NETBUFFERS parameter, 650, 652
- NETBUFSIZE parameter, 650, 652
- Netmask, configuration, 415
- netstat(TC), 433, 435, 436, 438, 447
- netstatus command, 642
- Network
  - activities, 436
  - buffer, 650
  - compatibility, 419
  - connections, 435
  - Control Block, 644, 656, 657
  - data files, 429
  - ID numbers, 419
  - kernel parameters, 930
  - planning, 15
  - printer, 162, 163
  - protocol, 408
  - sample, 403
  - server, 428
  - sysadmsh(ADM) selections, 35
  - system administrator responsibilities, 407
  - troubleshooting, 433
  - UUCP, 660
- Network address, for UUCP over TCP/IP, 692
- Network Information Center. *See* InterNIC
- Network Listing Service (NLS), 693
- Network Time Protocol. *See* NTP
- newgrp(C), 77
- "newproc - Process table overflow", 872
- NFILE parameter, 913
- NFS
  - adding user, 548
  - configuring, 549
  - daemons, 548
  - debugging, 554, 560
  - defined, 418
  - exporting a filesystem, 549
  - files, 547
  - handling remote filesystems, 562, 563
  - incompatibilities with NFS Release 1.1.1, 562
  - programs hung, 560
  - setting up clients, 551

## NFS (*continued*)

- setting up servers, 549
- starting, 554
- stopping, 554
- using with Automounter, 607, 612, 613
- using with NIS, 565
- NFSBUFS parameter, 656
- nfscast, error message, 632
- nfscnt(NADM), 556
- nfscd(NADM), 548, 555, 612
- nfs\_ghet, system call, 557
- nfsops, filesys(F) entry, 552
- nfsstat(NADM), 561
- NGROUPS parameter, 77, 915
- NHBUF parameter, 908
- NHINODE parameter, 912
- NIC. *See* InterNIC
- NINODE parameter, 912
- NIS
  - administering, 592
  - bind, 633
  - configuring, 573
  - daemons, 568
  - debugging, 596
  - defined, 565
  - domains, 567, 571, 573
  - initializing, 574
  - interoperability, 601
  - planning, 571
  - security, 570, 572, 596, 602
  - setting up servers, 573
  - starting, 581
  - user account, 569, 588, 589, 590, 591
  - using with Automounter, 611, 612, 617, 626, 633
  - using with NFS, 565
  - utilities, 569, 582, 588, 603
- NIST FIPS 151-1 conformance, 92
- NKDVTY parameter, 911
- NLBC parameter, 656
- NLBU parameter, 656
- NLOG parameter, 925
- NLS
  - ASCII terminal support, 312
  - Australia, 315
  - AUTOEXEC.BAT file, 310, 312, 313, 316, 318
  - background, 307, 308, 309, 310, 311, 312, 313, 314, 315
  - Belgium, 315

*NLS (continued)*

- Bourne shell, 314
- C shell, 314
- Canadian French code page, 309
- capabilities, 306
- CGA support, 312
- character conversion, 308, 312
- character sets, 306, 307, 310
- character translation, 306
- code page, 306, 308, 309, 310, 312, 315, 316, 317, 318, 319
- code set, 306, 308, 309, 310
- CODEPAGE environment variable, 314, 317, 319, 322
- CONFIG.SYS file, 310, 312, 313, 316, 318
- CONVOPTS environment variable, 324
- countries supported, 315
- country code, 315, 316
- COUNTRY environment variable, 314, 317, 319
- .cpi file, 309, 318
- defined, 305
- display considerations, 311
- display restrictions, 309
- DOS code page, 306
- DOS image, 312, 313, 316, 317, 318
- DOS Services setup procedures, 316, 317, 318, 320
- environment variables, 316
- /etc/cshrc, 314
- /etc/default/lang, 311, 319
- /etc/profile, 314, 317
- filename mapping, 308
- Finland, 315
- France, 315
- French, 310, 316, 317, 319
- Germany, 315
- Hercules support, 312
- ISO terminal setup procedures, 319
- ISO terminal support, 306, 313, 315, 319
- ISO\_TERM environment variable, 319
- Italy, 315
- KEYB environment variable, 314, 317, 319
- keyboard code, 315, 316
- KEYBOARD.SYS file, 318

*NLS (continued)*

- LANG environment variable, 310, 311, 314, 319, 322
- Latin America, 315
- MDA support, 312
- message files, 311
- MODE command, 318
- monitors, 309
- multilingual code page, 309, 316, 317
- Netherlands, 315
- NLSFUNC command, 316, 318
- Nordic code page, 309
- PC scancode terminal support, 312
- Portuguese code page, 309
- prerequisites, 306
- printers, 309
- .profile file, 314, 317, 319
- Spain, 315
- startup files, 312, 313, 316, 317
- supported displays, 306, 315, 316, 319
- Sweden, 315
- Swiss, 310
- Switzerland, 315
- territory, 310
- under DOS, 309, 310
- under DOS Services, 311, 312, 313, 314, 315
- under UNIX system, 310, 311
- United Kingdom, 315
- United States, 315
- UNIX code set, 306
- /usr/merge/image/std/vga, 317
- VGA setup procedures, 316, 317, 318
- VGA support, 306, 311, 315
- X Window System support, 306, 311, 315
- nladmin(ADM), with UUCP, 693
- NLSFUNC command, DOS, 306, 310, 316, 318
- NMOUNT parameter, 913
- NMPBUF parameter, 914
- NMPHEADBUF parameter, 914
- NMUXLINK parameter, 923
- NNCB parameter, 656
- NNCB\_NAMES parameter, 657
- "NO DEVICES AVAILABLE", 988
- "No more processes", 966
- No outgoing calls, UUCP messages, 994
- Node, 408
- NODE parameter, 922

NOFILES parameter, 913  
 nohup command, using with DOS processes, 340  
 nomodify flag, 534  
 nopeer flag, 534  
 noquery flag, 534  
 Nordic code page, 309  
 NOREAD option, Permissions file, 679  
 Normal operation, 45, 52  
 noserve flag, 534  
 "NOTICE: clalloc - Out of space on EAFS dev hd", 945  
 "NOTICE: s5ialloc - Out of inodes on EAFS /dev", 944  
 notrap flag, 534  
 notrust flag, 534  
 NOWRITE option, Permissions file, 679  
 NPBUF parameter, 908  
 NPROC parameter, 915  
 NPTE parameter, 657  
 NQUEUE parameter, 653, 922  
 NRECYCLE parameter, 657  
 NREGION parameter, 916  
 NSCRN parameter, 919  
 NSHINTR parameter, 926  
 NSPTTYS parameter, 919  
 NSTREAM parameter, 653, 922, 923  
 NSTREVENT parameter, 923  
 NSTRPUSH parameter, 923  
 NTIDS parameter, 657  
 NTP (Network Time Protocol)  
   add server dynamically, 544  
   algorithms, 529  
   configuration, 526, 527, 534  
   controlkey statement, 542  
   defined, 517, 521  
   delete server dynamically, 544  
   example configuration file, 530  
   guidelines, 524  
   keys file, 530  
   maxskew, 540  
   monitoring, 543  
   overview, 524  
   precision, 540  
   query, 541  
   requestkey statement, 542  
   subnet, 525, 546  
   syslog entries, 545  
   terms, 521, 522, 523  
   testing, 540

NTP (Network Time Protocol) (*continued*)  
   troubleshooting, 545  
   tuning, 540  
 NTP.MAXSKW, 540, 541  
 ntpq(ADMN), 541, 542  
 Null modem, 829  
 Null scope identifier, 638  
 NUMSP parameter, 925  
 NUMSXT parameter, 911  
 NUMTIM parameter, 653, 925  
 NUMTRW parameter, 653, 925  
 NUMXT parameter, 911  
 nuucp account, 691  
 NWB parameter, 657

## O

Object identifier  
   defined, 492  
   hierarchy, 493  
   types, 493  
 OK message, cu(C), 831, 959  
 On-card ROM, DOS images, 234, 237  
 "Open event driver failed", 966  
 Operating parameters, 646  
 Operating system  
   *See also* System  
   loading, 44  
   reinstalling, 932, 948  
 Operation modes, 45  
 Options file, dosenv.def, 263, 264, 266, 272, 273, 274, 275  
 Orange Book, 14  
 ORG domain name, 722  
 OS/2, 180  
 OSI (Open Systems Interconnection), 404  
 OTHER option, MACHINE entry, 682  
 "Out of clists", 872, 911  
 Out of network process table space (NPTE)  
   message, 657  
 Out of write behind table entries (NWB)  
   message, 657  
 Output command, 642, 643  
 Override terminal, 975

## P

### Packet

- NTP, 522
- trace, 433
- UUCP, 665, 689

### Page and swap usage, `sar -r`, 897

### Page size, printer, 133

### Paging activity, `sar -p`, 895

### Panic reboot, 55

### Parallel printer, 165

### Parameters

*See also* Kernel parameters

- audit, 37
- boot, 55
- filesharing, 651
- `lmxrc` file, 651
- login, 86
- MMDF, 742, 772, 782, 784-785
- modifying, 658
- network, 427
- operating, 641, 646
- UUCP, 689

### Parity values, modem problems, 961, 964

### Partition

*See also* DOS partition

- table, 181

### Partitioning hard disk

- assigning, 182
- DOS, 185
- `fdisk(ADM)`, 181
- installing UNIX on DOS machines, 184
- removing, 185
- two hard disks, 185

### `passmgmt(NADM)`, 569, 588, 590, 603

### `passwd` file. *See* `/etc/passwd` file

### `passwd(C)`, 419, 639

### Password

- assigning, 62-63, 72, 88
- changing, 57, 71-72, 956
- checking, 89
- compatibility with other UNIX systems, 79
- dial-in, 834
- disabling checks, 89
- expect string, 674
- expiration, 38, 72-73, 87-89
- matching and rejecting, 90
- root, 52
- security, 87
- super user, 12

### Password (*continued*)

- `sysadmsh(ADM)` options, 38
- Systems file, 662, 664
- UUCP logins, 668

### Path

- DOS, 237
- search, 243, 252

### Path separators, translating, under DOS Services, 300

### `PATH` variable, checking, 868

### Pathname, `()` command argument, 103

### `pathok`, error message, 630

### PC characters, 352, 353, 360, 367, 371

### PC keyboard simulation, 347

### PC scancode terminal

- input, 373
- output, 371
- setting up, 211
- supported by DOS Services, 285
- switchable, 211
- using, 344
- using with DOS, 353, 367, 371, 373

### PC scancodes, 352, 353, 360, 362, 367, 371

### `pcnfsd(NADM)`, 548

### PCT file, 360, 371

### `pcterm`, 365

- `terminfo` descriptor, 358

### Performance

- on client, 647
- tools, 881

### Permissions

- DOS partition, 222
- DOS Services, 221, 222
- locating files, 103-104
- mailbox file, 772
- problems, sticky bit setting, 955
- serial port, 696, 829
- `sysadmsh(ADM)` options, 41

### Permissions file, 662, 676

- combining entries, 682
- configuring, 677, 682
- errors, 706, 708
- example, 664, 667
- file transfer, 663
- format, 676, 677, 682
- granting access, 681-682
- options, 677

### Per-user forwarding, 822

### Phone field, Systems file, 673

## Phone line

- checking for noise, 961
- modem problems, 960, 992

pica pitch, 133, 158

ping(ADMN), 448

PIOMAP parameter, 910

PIOMAXSZ parameter, 910

Pitch, printer, 133

## Planning

- configuration, 571, 612
- installation, 14
- vtp, 639

PLOWBUFS parameter, 909

Plus symbol (+), in DOS Services commands, 281

Point-to-Point Protocol. *See* PPP

Poll, 521

Polling, setting for UUCP, 676

Populate existing, sysadmsh(ADM), 66

## Port

- COM. *See* COM port
- printer, 152-154
- problems, 968-970
- serial, 825, 829

portmap(NADM), 597, 600

Portmapper, 557, 558, 559

Portrait mode, 122

Portuguese code page, 309

postmaster, routing, 750

PostScript, options, 122

PostScript printer, 122

Power failure, recovering from, 863

Power supply, 15

Powering on, 44

## PPP (Point-to-Point Protocol)

- compared to SLIP, 453
- configuring, 454, 455
- defined, 441, 450
- Devices file entry, 458
- Dialers file entry, 458
- Ethernet/PPP gateway, 446
- installing, 450
- manual pages, 460
- removing, 458
- sharing serial line with UUCP, 450
- Systems file entry, 457
- Token-Ring/PPP gateway, 446
- troubleshooting, 458, 459, 460

Precedence definitions, 799

Precision oscillators, 532

precision statement, 540

Preprinted forms, 40

PRFMAX parameter, 911

Primary authorizations, 92

Primary server, 474, 522, 532

PRINT command, DOS, 299

## Print server

- defined, 507
- remote, setting up, 514

Print service, restricting user access, 132

Print streams, 214

Print wheel, 39, 125-128, 140

## Printer

- accepting, 39
- adding, 36, 39, 214
- administering, 131
- class, 138-139, 141
- configuring, 39, 120-122, 147-148, 165
- connection port, 125
- defaults, 39, 133, 139, 213
- dialup, 163
- disabling, 137, 215
- enabling, 137, 216
- fault, 129-131
- filters, 152
- fonts, 125-128
- forms, 139
- initializing, 165-166
- installing, 15, 122-128, 164
- interface program, 121, 156-162
- log files, 149-150
- modem connection, 125
- modes, 122
- network, 162, 163
- NLS support, 309
- output, sending to printer, 215
- parallel, 165
- rejecting, 39
- remote, 511, 513, 514, 515
- removing, 36, 39
- status, 35
- terminfo, 154-156
- terms, 120
- timeout, 216, 299
- troubleshooting, 967-974
- types, 122-125
- printer command, DOS Services, 214, 215, 216, 299, 321
- using to change printer timeout, 216



Printer port  
   adjusting, 152  
   attaching to DOS, 287  
   characteristics, 153-154  
   identifying, 215  
   initializing, 153  
 printerstat authorization, 92, 134  
 Printing  
   administering, 119-120, 141-150  
   attaching printers to DOS, 215  
   cancelling request, 137  
   command summary, 134-135  
   configuring, 150-163  
   daemons, 151  
   defaults, 145-147, 213, 214  
   DOS Services, 212, 213, 214, 215, 216  
   DOS, specifying printer, 215  
   effects of Ctrl-C keystroke, 299  
   filters, 39  
   forms, 39  
   in a multilanguage environment, 321  
   load management, 141  
   mail queue, 818  
   moving jobs, 143  
   priority, 143  
   process control, 214  
   queue, 40, 143-147  
   remote, 507, 508, 510  
   requests, 39, 119, 137  
   retry rate, 164  
   scheduler, 135  
   spooling, 119, 151-152, 165  
   starting, 135, 136, 142  
   stopping, 135, 136, 142  
   sysadmsh(ADM) options, 39, 41  
 printqueue authorization, 92  
 Printronix printer, 122  
 Priority, print job, 143  
 Privileges, sysadmsh(ADM) selection, 66, 74-75  
 Process  
   in loop, 967  
   memory management and swapping, 915  
   parameters, 915  
   regions, 916  
   scheduling, 41  
   specifying maximum time slice, 918  
   sysadmsh(ADM) options, 40  
   troubleshooting, 966-967, 974-975  
 Process accounting, 902

Process environment table, 657  
 Process table, allocating entries, 915  
 .profile file  
   in a DOS Services NLS environment, 314, 317, 319  
   UNIX, 243  
 .profile file, UNIX  
   setting DOS environment variables in, 338  
   specifying DOS as default shell, 339  
 Program  
   simultaneous execution, 13  
   system specific, 37  
 Prompt, login, 52  
 PROMPT environment variable, how to set, 338  
 Propagating NIS maps, 567, 584, 587  
 Protected Mode, 244, 334  
 Protected password database, 58, 80  
 Protection, DOS, 222  
 Protocol, 711  
   BGP, 483  
   communications, 408, 409  
   defined, 403  
   EGP, 483  
   exterior gateway, 483  
   HELLO, 483  
   interior gateway, 483  
   OSI model, 404  
   RIP, 483  
   set, 648  
   SNMP, 491, 492  
   statistics display, 438  
   TCP/IP, 409  
 prwarn(C), 74, 88  
 ps(C), checking process activity, 888  
 Pseudo-ttys  
   administering, 432, 433  
   default, 919  
 pstat(C), 899, 903  
 Public ftp account, 417  
 putbuf buffer, 910  
 PUTBUFSZ parameter, 910  
 pwconv(ADM), 58  
 pw\_id\_map file, 50

## Q

QIC-02 tape drive size, 169  
 Query NTP, 541  
 queryspace authorization, 92  
 Queue  
   forcing, 813  
   interval, 813  
   intervals in sendmail, 815  
   MMDF, 770  
   printer, 40, 119, 143-147  
   priorities in sendmail, 815  
   runs, forking, 815  
 Queued messages, 795  
 Quitting  
   login session, 13  
   sysadmsh(ADM), 42  
 Qume Sprint 1155 printer, 122  
 quot(C), 102

## R

RAM buffer size, configuring, 412  
 rcfsck, filesystems(F) entry, 552  
 rcmount, filesystems(F) entry, 552  
 read ahead parameter, 647, 648, 649  
 READ option, Permissions file, 678  
 Read timeouts in sendmail, 815  
 reada timeout parameter, 648  
 reado parameter, 648  
 Read-only device access, specifying, 289  
 Reboot. *See* Boot  
 Rebuilding the alias database, 821  
 Records, locked by system, 914  
 Recursion, 103-105  
 Redrawing, screen, 350  
 Region table, 916  
 reject(ADM), 142  
 Rejecting, printer requests, 142  
 Relative pathnames, 171  
 Relaxed security, 35, 83  
 Remote  
   command execution, 659, 665  
   computer link, 663  
   drives, interpreting as local, 255, 256  
   files, interpreting as local, 256  
   filesystem mounting, 551  
   mount failed, 556  
   name server, 463, 466  
   printer, 511, 513, 514, 515

Remote (*continued*)  
   printing, 507, 508, 510  
   UUCP connection, 665  
 remote.unknown script, 689  
 remount, error message, 634  
 Removing  
   clients, 640  
   DOS applications, 252  
   DOS option assigned values, 302, 303  
   installed options, from dosenv.def file, 275  
   NIS servers, 593  
   printer class, 141  
   remote printing, 508  
   user, 57, 67  
   user accounts, 591  
 Request ID, printing, 119, 137  
 Request log, printer, 149  
 REQUEST option, Permissions file, 677  
 requestkey statement, 542  
 Request/reply transactions, 642  
 Reset, audit rules, 37  
 Resource records, 467, 469, 470, 471, 472, 473  
 Resources  
   *See also* System resources  
   X, 328, 329  
 Restart system, 37  
 RESTORE command, DOS, 217  
 Restoring  
   filesystems, 865, 945, 948  
   installed option files, 275  
   space, 946  
   system files, 932-938  
 restrict statement, 534  
 Restricting file access, 417  
 Retire, sysadmsh(ADM) selection, 67  
 Retries establishing a session, 656  
 Retry period, UUCP, 672  
 Return key, 12  
 Return-Receipt-To header line, 822  
 Rewriting rules, 797, 801, 802, 805  
 RFC, gateway protocols, 489  
 .rhosts, 429  
 RLP, 507  
 rlpconf(ADMN), 513, 514  
 rlpstat(C), 510  
 rmail(ADM), 677, 680, 682  
 rm(C), command, 252  
 rmtab(NF), 555  
 rmuser(ADM), 67  
 ROM, DOS images, 234, 237

**Root***See also* Super user

authorization, 78

boot, 45

login, 52

password, 52

security, 65, 975

**Root filesystem, 95**

mount table, 913

restoring, 948

**Root floppy**

Emergency Boot Floppy Set, 932, 938

restoring /etc/emulator file, 934

restoring /etc/init file, 938

restoring root filesystem, 948

**root.cache, 475****Roundtrip delay, 522****routed(ADMN), 446, 483, 484, 485****Routing**

default, 484

files, converting, 763

MMDF, 670

table, 416, 437

wildcard, 484

**rpcinfo(NADM), 555, 556, 558****RS-232**

connecting the cable, 696

interface, 119

selecting a serial port, 695

UUCP requirement, 661

**Rule sets, rewriting, 803****Runaway process, 967**

killing, 974

**Running**

AUTOEXEC.BAT file, 298

DOS application programs, in background, 340

games, under DOS Services, 297

**S****S5CACHEENTS parameter, 915****S5HASHQS parameter, 915****"s5iread - ... inode table overflow", 872****S5OFBIAS parameter, 915****S86mmdf file, 767****sadc (System Activity Data Collector)**

command, 889

**Safe to Power Off message, 53****Safety, 15**

sar(ADM), 889-900

Scan window, 30, 35

Scancode terminal. *See* PC scancode terminal

scancodes, 352, 353, 360, 362, 367

Schedule field, Systems file, 671

**Scheduler**

printing, 135

problems, 941, 967

"scheduler is not running", 967

**Scheduling commands, UNIX**

precautions when using with DOS, 340

using with DOS, 340

scoedit(X) performance, 880

scohelp(X) performance, 880

scologin(X), performance, 880

scologin(Xm) does not start, 966

scomail(X) performance, 880

scopald(X), 880

Scope identifier, 638

scosession(X) performance, 880

scoterm(X) performance, 880

**Screen**

shifting, 349

switch, 209, 210

Screen saver, TBLNK parameter, 918

Screens, multiple, 13

SCRNMEM parameter, 919

SCSI, bad block handling, 943

scsibadblk, 943

SDSKOUT parameter, 927

Search. *See* find(C)

Search path, 243, 252

SECCLEARID parameter, 927

SECLUID parameter, 927

Secondary, server, 474, 522, 532

Secondary authorizations, 92

SECSTOPIO parameter, 927

**Security**

authorizations, 77, 90

C2, 14

crash recovery, 866

database, 47, 58-61, 866, 975

defaults, 81, 83

disabling C2 features, 927

DOS partition protection, 220, 221, 222

failure mode, 83

features, 48-50, 866, 886, 975

file permissions, 222

high, 83

### Security (*continued*)

- improved, 83
- kernel parameters, 927
- low, 83, 93
- NIS, 570, 572, 596, 602
- parameters, 81, 83
- password, 87
- preventing access to devices, 290
- problems, 887
- reports, 38
- retirement, 83
- specifying read-only device access, 289
- super user, 12
- sysadmsh(ADM) options, 35, 38
- traditional, 93
- troubleshooting, 975-979
- UID reuse, 83
- UNIX system, 83
- UUCP, 680-681

Selecting, Configuration menu, dosadmin, 267

SEMAEM parameter, 921

Semaphores, 920-921

SEMMAP parameter, 920

SEMMNI parameter, 920

SEMMNS parameter, 921

SEMMNU parameter, 920

SEMMSL parameter, 920

SEMOPM parameter, 920

SEMUME parameter, 920

SEMMVMX parameter, 921

Send string, login script, 673

SENDFILES option, Permissions file, 678

sendmail

- administering, 817
- alias database, 817
- arguments, 813
- changing option values, 814
- collecting messages, 792
- command line flags, 810
- compared to delivermail, 790
- compared to MMDF, 790
- compared to MPM, 791
- configuration file, 796, 799, 804, 814, 820
- daemon mode, 813
- debugging, 813
- delivering messages, 793
- delivery mode, 816
- editing the message header, 794
- error mailer, 804

### sendmail (*continued*)

- file modes, 816
- flags, 803
- mail queue, 818
- mailer flags, 803
- Message Processing Module, 791
- options, changing values of, 814
- queue, 813
- return to sender, 793
- rewriting rules, 797
- special header lines, 822
- suid, 816
- support files, 822
- temporary file modes, 817
- tuning, 814, 815, 816

Sequence, escape, 354

Serial cable, 696

Serial card

- multiport, 331, 826
- sysadmsh(ADM) selection, 36

Serial line

- See also* SLIP
- conflicts, 834
- connecting, 15
- lock file, 664
- settings, standard, 964
- UUCP requirement, 661

Serial port

- choosing, 825
- permissions, 696, 829
- UUCP connection, 695-696

Serial terminal, 285

- keyboard mapping, 345, 347, 348
- limitations, 345
- setting up, 211
- TERM setting, 348
- using with DOS, 352, 353, 360

serial(HW), 829

Server

- defined, 402
- displaying name, 642
- indicator, 657
- kernel data space, 657
- mode, 522
- NFS, 549, 550, 560, 631
- NIS, 567, 572, 573, 592, 593, 594
- types, 462, 463

Server system, kernel configuration, 902

- Session
  - establishing, 656
  - status table, 644
- setany(ADMN), 502, 503
- sg(C), 77
- SGID bits, 927
- Shadow inodes, 110
- Shared
  - data parameters, 921
  - DOS/UNIX applications, 242-248
  - DOS/UNIX filesystem, configuring the DOS environment, 255, 256
  - DOS/UNIX filesystem, DOS interrupts, 332
  - filesystem, backup procedures, 217
  - libraries, specifying maximum number, 914
- Sharing, disk drives, 218
- Shell
  - escape, sysadmsh(ADM), 34
  - exiting, 13
  - layers, 36
  - login, 65
- Shell, UNIX, setting DOS environment
  - variables from, 337, 338
- SHLBMX parameter, 914
- SHMALL parameter, 921
- SHMMAX parameter, 921
- SHMMIN parameter, 921
- SHMMNI parameter, 921
- SHMSEG parameter, 921
- showmount(NADM), 555, 559
- Shutdown
  - abnormal, 863, 951
  - menu option, 37
  - procedure, 863
- shutdown(ADM)
  - authorization, 92
  - procedure, 52
  - single-user option, 115
  - sysadmsh utility, 46
  - system maintenance mode, 46
  - X-window execution, 46
- SIGHUP signal, 161
- SIGINT signal, 161
- SIGPIPE signal, 161
- SIGQUIT signal, 161
- simple content type, 124
- Simple Network Management Protocol. *See* SNMP
- Single-user mode, 45
  - cannot enter, 937
  - changing kernel parameters, 882
  - fixing a bad track, 943
  - restoring /unix, 934
  - shutdown(ADM), 114, 115
- sio file, 830
- Site planning, 14
- Sitename, verifying unique, 994
- Size, displaying, 102
- Skew, NTP, 522
- Slash (/), root filesystem, 95
- Slash character (/), 300
- slattach(ADMN), 445
- Slave server, NIS, 567
- Slew, NTP, 522
- SLIP (Serial Line Internet Protocol)
  - See also* Serial line
  - compared to PPP, 453
  - configuring, 415, 442, 444
  - defined, 441
  - dialup connection, 444
  - direct connection, 442
  - error messages, 448, 449
  - Ethernet/SLIP gateway, 446
  - installing, 441
  - manual pages, 450
  - removing, 447
  - Token-Ring/SLIP gateway, 446
  - troubleshooting, 445, 447, 448, 449
- slot(ADM), 54
- Smartmodem
  - 1200 or compatible, 826
  - 2400, V-series 9600, 827
- SMB protocol, virtual circuit dialect, 642
- smnck script, 48
- SMTP, MMDF, 758
- SMUX (SNMP Multiplexing), 494
- Snapshots, 233
- SNMP (Simple Network Management Protocol)
  - agents, 495
  - authentication, 495
  - basic concepts, 491
  - commands, 497
  - communities, 495
  - configuration, 496, 497, 498
  - daemon, 496
  - defined, 491
  - Management Information Base (MIB), 494

SNMP (Simple Network Management Protocol) (*continued*)

- management stations, 495
- object identifier, 492, 493
- protocol defined, 491, 492
- relevant RFC's, 506
- sample file, 496, 497
- SMUX protocol and, 494
- Structure of Management Information (SMI), 492
- traps, 495
- using, 499-506
- snmpd sample, 496, 497
- snmpstat (ADMN), 501, 502
- SOA (Start of Authority) record, 469
- SO\_DEBUG option, 433
- Soft mount, 551, 555
- SORT command, DOS, 291
- "sp: spclose - Cannot allocate STREAMS block", 872, 923
- space.c files, TCP/IP, 930
- Special
  - action escape sequences, 364
  - classes, 801
  - device file, 696
  - device node, creating, 973
  - macros, 799
- Specifications, device, 229
- Speed conversion, 839
- Speed field, Systems file, 672
- Spooler
  - printer, 119, 165
  - UUCP, 660
- SPTMAP parameter, 916
- SRI International, 723
- srvdaemon, virtual terminal support, 639
- SRVINUSE parameter, 657
- Stack region, 916
- STACKS command, DOS, 235
- "Stage 1 boot failure: error loading /boot", 932
- Standard resource record format, 467
- Standard ROM, DOS images, 234, 237
- standard terminals, output, 355
- Starting
  - DOS, 250
  - DOS environment, without DOS image, 296
  - NFS, 554
  - NIS, 581
  - print service, 135, 136, 142

Starting (*continued*)

- system, 43
- terminal output, 13
- UUCP network, 660-661
- Startup
  - automatic, 641
  - mounting filesystems, 97
- Startup memory, acceptable values for, 270
- startx(Xm) does not start, 966
- statfs(S), system call, 557
- Statistics, audit, 37
- STATUS error messages, 707
- .Status file, error, 707
- Status files, UUCP, 991, 993
- Status Line (sysadmsh), 19
- stdat structures, 653
- Step, NTP, 522
- Sticky bit setting, unable to remove files, 955
- stopio(S), 927
- Stopping
  - client, 641
  - DOS processes, 349
  - login session, 13
  - NFS, 554
  - print request, 137, 142
  - print service, 135
  - program, 12-13
  - sysadmsh(ADM), 42
  - system, 37, 52
  - terminal output, 13
- Storing files, 40, 167, 170, 173, 175
- Straight-through cables, 829
- Stratum, NTP, 522
- STRCTLSZ parameter, 924
- STREAM-head structures, 923
- Stream-oriented DOS programs
  - definition, 291
  - specifying, 291
  - test for verifying, 292
- STREAMS, 922
  - parameters, 922
  - sysadmsh(ADM) option, 36
  - tuning, 433
  - viewing crash(ADM) usage statistics, 884
- Streams, 650, 651, 652, 653, 656
- "string: Configuration buffer full", 872, 911
- STRLOFRAC parameter, 924
- STRMEDFRAC parameter, 925
- STRMSGSZ parameter, 924
- "stropen1 - Out of streams", 872, 923

"stropen2 - Out of streams", 872  
 Structure of Management Information (SMI), 492  
 stty(C)  
     adjusting printer port characteristics, 153  
     default options, 153  
     printer port characteristics, 153-154, 159  
 stune(F), 646, 652, 907  
 Subdomain, 722  
 submit(ADM), 715  
 Subnetwork, 427  
 SUBST command, DOS, 241, 242  
 Subsystem  
     audit, 93  
     authorizations, 91-92  
     database boot message, 47  
 su(C)  
     accessing other accounts, 78  
     authorization, 78  
     security, 65  
 SUID bits, 927  
 suid in sendmail, 816  
 sulog file, 108  
 Super block, 113  
     correcting values, 953  
     repairing damage with fsdb(ADM), 951  
 Super user, 12  
     *See also* Root  
     exiting, 52  
     login, 52  
     security, 12, 52, 65  
     single-user mode, 45-46  
 Superkey, 333  
 .suppgroups file, 77  
 Support files, summary of, 822  
 suspendaudit authorization, 92  
 "swapdel - Total swap area too small", 872, 917  
 Swapping  
     memory, 915  
     parameters, 915  
     reporting activity, 899  
 Switch, settings, modem, 827  
 Switch characters, translating, under DOS Services, 300  
 Switch screens, 209, 210  
 Switchable scancode terminals, examples of, 211  
 SWITCHES command, DOS, 235

Switching  
     activity, reporting, 899  
     operating systems, 182-183  
     screens, 13  
 switchkey command, DOS Services, 209, 210  
 Symbol table, size, 912  
 Symbolic links, 607, 611, 629  
 Symmetric active mode, 522, 533  
 Symmetric passive mode, 522  
 sync(ADM), 114-115  
 Synchronization subnet, 523  
 Synchronizing clocks, 517  
 Syntax  
     dos command, 262, 281, 282, 283  
     dosdev file, 231, 232  
     dosopt command, 272, 274, 281, 282, 283  
     merge command, 253  
     specifying DOS options, 262, 263  
 SYS command, DOS, 332  
 sysadmin authorization, 91  
 sysadmsh(ADM), 11, 17  
     backups, 945  
     color selection, 34  
     configuring printer, 213  
     creating accounts, 569, 588, 589  
     customizing menus, 34, 36  
     error messages, 20  
     filesystem options, 96-97  
     keystrokes, 24-27  
     menu, 11, 19-28, 35  
     mount filesystem, 945  
     options, 19-22, 36-37  
     printer options, 121, 123, 125-126  
     quitting, 42  
     restoring filesystem, 945  
     shell escape, 34  
     shortcuts, 35  
     system shutdown, 46  
     unmount filesystem, 945  
     user ID, 419, 639  
     user selection, 21  
 Sysfiles file, 688  
 syslog entries, NTP, 545  
 sys.precision, 540, 541  
 System  
     backup, 217  
     buffer cache, tuning, 872  
     call activity, sar(ADM) -c, 892  
     checking status of tables, 898  
     cleaning, 44

## System (continued)

- crash, 866
- defaults, 36
- defaults database, 59
- diagnosing inefficiency, 881
- equivalence, 417, 429
- file, 238
- files, 932-938, 946
- hardware, 53-54
- increasing reliability, 910
- log, 860
- maintenance, 9, 12, 100, 106-107
- maintenance mode, 45-46
- name, 411, 761, 922
- networking requirements, 419
- panic, 864-865
- planning, 14
- safety, 15
- security, 48-50, 866, 886, 975
- shutdown, 37
- shutdown, abnormal, 863
- starting, 43
- stopping, 37, 46, 52
- time, 46
- System Activity Reporter. *See* sar(ADM)
- System activity, sysadmsh(ADM), 35
- System administrator, 9, 10
  - DOS, 206
  - duties, 407
  - log book, 10
  - super user login, 12
  - tools, 11
  - user maintenance, 57
- "SYSTEM NOT IN Systems FILE", 987
- System resources
  - buffers, 905
  - kernel, 882, 887
  - limits, 905
- systemid file, 640
- Systems file
  - baud rate, 666
  - device requirements, 663
  - error, 708-709
  - error with cu(C) command, 987
  - examples, 664, 667
  - for TCP/IP, 691
  - format, 670
  - Local Area Network, 673
  - login sequence, 662, 664
  - phone field, 673

## Systems file (continued)

- PPP entry, 457
- remote computer link, 663
- schedule field, 671
- security, 670
- speed field, 666, 672
- TCP/IP address example, 694
- testing modem, 669
- UUCP, 662, 992-993
- verifying phone number, 669

## T

- t files, 48-49
- Table entries, 905
- Table overflow, 871-872
- tail(C), 107
- Tailoring DOS Environment menu, 270
- Tandy DMP printers, 122
- Tape
  - archiving files, 170
  - buffers, values, 909
  - /etc/default files, 169
  - sysadmsh(ADM) options, 40
  - troubleshooting, 979-982
- Tape drive, 167
  - DAT, 170
  - default device, 168
  - Error Correction Code (ECC), 172
  - /etc/default files, 169
  - Exabyte (8mm), 170
  - Irwin, 172
  - maintaining, 171
  - mini-cartridge, 170
  - performance, 909
  - QIC-02, 169
  - sizes, 169
  - sysadmsh(ADM) selection, 36
- tape(C), 171
- tar(C)
  - archiving files, 170, 176
  - cannot open /dev/rct0, 981
  - default settings, 169
  - directories, 170
  - pathname limit, 171
  - relative versus absolute pathnames, 171
- TBLNK parameter, 918
- tcb, boot message, 47
- /tcb/audittmp files, 107



- tcback(ADM), 47-50, 866, 978
- /tcb/files/auth directory, 58
- TCP
  - dialer type, 684
  - reliable transmission, 409
  - socket interface, with UUCP, 690
- TCP/IP
  - defined, 408
  - end user commands, 410
  - NetBIOS, 636, 638
  - printer configuration, 162, 163
  - protocol, 405, 408, 409
- TCSEC (Trusted Computer System Evaluation Criteria), 81
- Telebit modem, 828
- Temporary
  - file modes, sendmail, 817
  - files, locating, 104
  - files, removing, 100, 105, 106
- TERM environment variable
  - printing, 126, 154-155
  - setting, 211
  - terminal type, 984
- Terminal
  - administering, 38
  - authorization, 91
  - control entry, cannot rewrite, 956
  - disabled, 349, 979
  - display, 13
  - display adapters used, 286
  - enabling, 36, 979
  - input, 362
  - I/O, sar -y, 900
  - locking, 69
  - NLS sample setup for ISO, 319
  - override, 975
  - scancode. *See* PC scancode terminal, 345
  - serial. *See* Serial terminal, 345
  - standard vs PC scancode, 367
  - supported by DOS Services, 285, 341
  - sysadmsh(ADM) options, 36, 38
  - troubleshooting, 982-985
  - unlocking, 69
  - using with DOS, 351, 353, 360, 362
  - "Terminal is disabled -- see Account Administrator", 979, 985
  - "Terminal is disabled but root login is allowed", 977
- Terminals, standard vs PC scancode, 353
- terminfo(F), 124, 154-156
  - defined, 352
  - files, 352
  - standard terminal output, 355
- terminfo(M), 351
- Testing
  - MMDF, 769-770
  - NTP, 540
- Texas Instruments 855 printer, 122
- Text files, converting, 322
- Text region, 916
- "The security databases are corrupt", 976
- "There may be a system dump memory image in the swap device", 860
- Time
  - client, 523
  - daemon, 523
  - server, 523
  - setting, 46
  - synchronizing, 517
  - sysadmsh(ADM) selection, 36
  - used by command, 901
- Timecode
  - receivers, 532
  - transmitters, 532
- timed(ADMN), 517, 520
- timedc(ADMN), 520
- TIMEOUT, 55
- "timeout - Timeout table overflow", 872, 918
- Timeouts in sendmail, 814
- timex(ADM), 901
- TIMEZONE parameter, 918
- TLI
  - dialer type, 684
  - NUMTRW parameter, 925
  - read/write data structures, 653
  - UUCP, 692
- TLIS dialer type, 684
- TM. (temporary data) files, 665, 702
- /tmp directory, 100
- /tmp\_mnt, 607, 615, 629
- Token-Ring/SLIP gateway, 446
- Top-level domain, 721
- Trailblazer modem, 828
- Transport Layer Interface. *See* TLI
- Traps, 495, 523
- TREE command, DOS, 291
- Tree connects, 657

Troubleshooting

- DOS Services, 333
- MMDF, 770
- NetBIOS level, 643
- network, 433
- NTP, 545
- PPP, 458, 459, 460
- SLIP, 445, 447, 448, 449
- SNMP, 504, 505, 506
  - using SNMP, 491
- utilities, UUCP, 996
- trpt(ADMN), 433
- truechimer, 529
- Truncation, filename, 99
- Trusted
  - access, 640
  - alias table, 782
  - Computing Base, 977
  - system, 14, 88
  - users, 799
- trymany, error message, 631
- tset command, reinitializing terminal, 983-984
- TSP (Time Synchronization Protocol), 517, 518, 519, 520
- ttys
  - database, 978
  - modem, 826
  - root, 975
- ttyupd(ADM), 51
- Tunable kernel parameters. *See* Kernel parameters
- Tuning
  - NTP, 540
  - sendmail, 814, 815, 816
  - STREAMS, 433
- TYPE command, DOS, 292
- TZ (timezone) variable, 918

## U

- UC\_SMBPATH, 652
- ULIMIT parameter, 916
- umnt(C), 553
- umount(ADM)
  - floppy filesystem, 933-934
  - hard disk root filesystem, 935
  - use of, 97, 114, 178
- umount(NADM), 547, 627

- uname(C), 640
- Undo structures, number in system, 920
- Unique, sitenames, UUCP, 994
- UNIX
  - /dev/hd0a, device name, 181
  - filesystem, repairing with fsdb(ADM), 954
  - keyboard, 12
  - partition on DOS systems, 184
  - removing partition, 185
  - /unix, restoring, 933
- UNIX code set, 306, 308, 309, 310
- UNIX commands, using scheduling commands with DOS, 340
- UNIX message files, 311
- "unix not found", 933
- UNIX print spooler
  - bypassing, for DOS printing, 215
  - changing printing timeout, 216
- UNIX shell
  - control code for forking, 349
  - default, 339
  - setting DOS environment variables from, 337, 338
- unix2dos command, DOS Services, 306, 322
- /unix.old, alternate kernel file, 54, 933
- Unkillable process, removing, 975
- Unknown, printer type, 123
- "UNKNOWN FILE SYSTEM VERSION", 948
- "Unknown Host Domain", 958
- Unlocking
  - account, 976
  - console keyboard, 939-941
  - terminal, 57, 985
  - user account, 68, 956
- Unmounting, print wheel, 141
- "usemouse: can't open slave (13)", 965
- "usemouse: no mouse available(5)", 965
- User
  - accounts, managing, 548
    - adding, 57, 60-63, 66, 548, 569, 588, 589, 590
  - administrative, 66
  - auditing, 75
  - authorizations, 74-75
  - disk usage, 102
  - DOS Services, 206
  - equivalence, 417, 429
  - group, 69
  - home directory, 66
  - locking account, 68
  - modifying accounts, 64-76, 81, 83, 591

*User (continued)*

- printer access, restricting, 132
- removing, 57, 67
- removing accounts, 591
- retiring, 104
- type, 65
- unretiring, 67

*User ID*

- consistency, 639
- login field, 65
- network, 419
- selecting, 548
- sysadmsh(ADM) selection, 419

"*useshell*: File Control database inconsistency", 62, 979

"*useshell*: Script path

/usr/lib/mkuser/mkuser.init may be compromised ...", 979

/usr/adm/messages file, 107

/usr/adm/pacct file, 107

/usr/adm/sulog file, 106

/usr/adm/syslog, 545

/usr/bin directory, 661

/usr/include/sys/netinet/in.h, 692

/usr/include/sys/socket.h, 692

/usr/lib/event/ttys, 966

/usr/lib/goodpw/checks directory, 89

/usr/lib/goodpw/match file, 90

/usr/lib/goodpw/reject file, 90

/usr/lib/lm, 646

/usr/lib/terminfo file, printer types, 971

/usr/lib/uucp, 661-662, 665

/usr/lib/uucp/uucico, login shell, 668

/usr/merge/image/std/vga, use with NLS, 317

/usr/mmdf/chans, 757

/usr/mmdf/log, 770

/usr/mmdf/mmdftailor, 739

/usr/mmdf/table directory, errors, 957

/usr/net/nls/inet/tcp directory, 693

/usr/spool/lp/logs/request files, 107

/usr/spool/lp/model file, 156

/usr/spool/mail, 770, 959

/usr/spool/mmdf, 770

/usr/spool/uucp, 662, 991

/usr/spool/uucp/.Log directory, 991

/usr/spool/uucp/LOGFILE file, 107

/usr/spool/uucp/.Log/.Old files, 107

/usr/spool/uucppublic, 662

/usr/spool/uucp/.Status directory, 991

*Utilities*

- accessing DOS, 186
- administering, 645
- NIS, 569, 582, 588, 603
- vtp, 645

uucheck(ADM), 661, 994, 996

uucico(ADM), 991

uucico(C), 660, 662-665, 689

uuclean(ADM), 661

UUCP, 659-660

access, 662

ACU (Automatic Call Unit), 672, 684

administering, 701

configuration files, 36

configuring, 665, 670-671

connecting, 660-661

creating login accounts, 667

daemons, 661-665, 681-682, 689

default paths, 680

defined, 421, 422

dialer program, 445

error messages, 704-705, 707

examples, 697

granting access, 680-682

leased line, 667

linking, 684

Local Area Network (LAN), 683-684

login, 667-668, 670, 673-674

messages, 991-994

MMDF, 722, 758

modem, 661, 665, 666, 668, 829

node name, 670

over TCP/IP, 684, 689

password, 668

polling, 676

port, 829

printer configuration, 162, 163

protocols, 688

sample transaction, 664

security, 662, 670, 672, 676, 679-681, 689

setting up, 661

sharing serial line with PPP, 450

status, sysadmsh(ADM) selection, 35

troubleshooting, 974, 985-996

*UUCP commands*

permissions problems, 988

remote execution, 665

uucheck(ADM), 996

uucico(ADM), 991

uulog(C), 989

UUCP commands (*continued*)

uustat(C), 989  
uutry(ADM), 990

UUCP directories

public directory, 677  
security, 679  
spool directory, 702  
/usr/bin/uucp, 662  
/usr/spool/uucp, 662  
/usr/spool/uucppublic, 662

"uucp failed ...", 988

UUCP files

C. (work) files, 664, 702-703, 705-706  
control files, 660, 664-666  
D. (data) files, 702, 704, 706  
database control files, 660  
Devices file, 662, 666, 683, 684  
Dialcodes file, 688  
Dialers file, 664, 685-688, 709, 835  
editing, 666  
LCK (lock) files, 664, 669, 702-703, 705, 709  
.Log directory, 991  
log files, 107, 661, 662, 665, 701-702  
Permissions file, 662-663, 677  
permissions problems, 994  
routing files, converting, 763  
.Status directory, 991  
Sysfiles file, 688  
Systems file, 670-673  
TM. (temporary data) files, 665, 702  
uudemon.hour, 663  
uulog(C), 702  
X. (execute) files, 661-663, 665, 702, 704-705

uucp(C), 659, 662, 664, 835, 988-990

uucppublic directory, 678

uudemon.admin script, 701

uudemon.clean script, 107, 661, 701-702

uudemon.hour script, 663, 701

uudemon.poll script, 701

uinstall(ADM), 660

uulist(ADM), 764

uulog(C), 661, 702, 989, 996

UUname, 742

uuname(C), 701, 764, 996

uusched(ADM), 663-664

uustat(C), 661, 989, 996

uuto(C), 662, 702

uutry(ADM), 669, 674, 709, 990, 996

uux(C), 659, 680

location, 662

problems, 996

records, 662, 702

specifying options, 766

UUCP channel relationship, 758

uuxqt(ADM), 681

executing remote programs, 663, 665

MACHINE option, 682

querying, 702

X. (execute) files, 663

## V

v.42bis, 963

VALIDATE entry, Permissions file, 681

Ventilation, 15

## VGA

DOS Services, 209

in an DOS Services NLS environment, 306, 315, 316

## VGA card

requirements, 344

UNIX MultiScreen functionality, 344

VGA\_PLASMA parameter, 926

vhand daemon, 916

Viewing, file, 41

## Virtual

diskette, 223

DOS floppies, 226

partition, 223, 224

terminals, 911, 919

## Virtual circuit

dormant, 657, 658

for client, 652

ID number, 642

request/reply transactions, 642

state, 642

Virtual machine, booting, 297

vmstat(C), 893, 896, 899, 903

Volume, modem, 828

vtp, 639, 645

## W

wall(ADM), 101

WAN (Wide Area Network), 402

"wd: ERROR on fixed disk ...", 943

Well known services resource record, 471

Well-behaved DOS programs. *See* stream-oriented DOS programs  
 Wheel, print, 125, 127-128  
 Wide area network. *See* WAN  
 Window, scan window, 30  
 Wiring, modems, 829  
 Write  
     requests, 649  
     window, 647, 649  
 WRITE option, Permissions file, 678  
 writeaudit authorization, 92  
 Write-behind structures, 657  
 write(C), 128

## X

X client setting, DOS, 302  
 X. (execute) files, 661-663, 665, 702, 704-705  
 X resources, 328, 329  
 X server, error messages, 966  
 X Window System, in an DOS Services NLS environment, 306  
 X.25, protocol, 404  
 xargs(C), 105  
 xclock(X) performance, 880  
 xdt3(X) performance, 880  
 XENIX  
     filesystem, repairing with fsdb(ADM), 954  
     MMDF, 763  
     shared data segments, 921  
     special semaphores, 921  
 XITONCLOSE parameter, 658  
 XMERGE environment variable, 330  
 XNETINODE parameter, 658  
 xntpd(ADMN), 541, 542  
 xntpres program, 536  
 XON/XOFF, printer, 129  
 "xsd\_alloc - XENIX semaphore table overflow", 872  
 "xsd\_alloc - XENIX shared data table overflow", 921  
 XSDSEGS parameter, 921  
 XSDSLOTS parameter, 921  
 "xsem\_alloc - XENIX semaphore table overflow", 921  
 "xsemfork - File table overflow", 914  
 XSEMMAX parameter, 921  
 xterm(X) performance, 880

"xtinit - Cannot allocate xt link buffers", 872, 911

## Y

Yellow Pages. *See* NIS  
 "You do not have authorization to run ...", 979  
 ypbind(NADM), 568, 598, 600, 603  
 ypcat(NC), 592, 593, 603  
 ypcnt(NS), 603  
 ypinit(NADM), 567, 573, 574, 582, 583, 603  
 ypmake(NADM), 582, 583, 589, 594, 604  
 ypmmaplate(NF), 570, 573, 604  
 ypmatch(NC), 604  
 yppassmgmt(NADM), 569, 588, 589, 591, 604  
 yppasswdd(NADM), 568, 602, 604  
 yppasswd(NC), 602, 604  
 yppoll(NADM), 604  
 yppush(NADM), 585, 604  
 ypservers, NIS map, 585  
 ypserv(NADM), 568, 585, 597, 598, 604  
 ypset(NADM), 605  
 ypwhich(NADM), 599, 605  
 ypxfrd(NADM), 568, 605  
 ypxfr(NADM), 584, 585, 587, 596, 605

## Z

"Zero length FP emulator file", 934



Please help us to write computer manuals that meet your needs by completing this form. Please post the completed form to the Publications Manager nearest you: The Santa Cruz Operation, Ltd., Croxley Centre, Hatters Lane, Watford WD1 8YN, United Kingdom; The Santa Cruz Operation, Inc., 400 Encinal Street, P.O. Box 1900, Santa Cruz, California 95061, USA or SCO Canada, Inc., 130 Bloor Street West, 10th Floor, Toronto, Ontario, Canada M5S 1N5.

Volume title: \_\_\_\_\_  
(Copy this from the title page of the manual)

Product: \_\_\_\_\_  
(for example, SCO UNIX System V Release 3.2 Operating System Version 4.0)

How long have you used this product?

- ☐ Less than one month      ☐ Less than six months      ☐ Less than one year  
☐ 1 to 2 years      ☐ More than 2 years

How much have you read of this manual?

- ☐ Entire manual      ☐ Specific chapters      ☐ Used only for reference

|                                                                                               | Agree                    |                          | Disagree                 |                          |
|-----------------------------------------------------------------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| The software was fully and accurately described                                               | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| The manual was well organized                                                                 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| The writing was at an appropriate technical level<br>(neither too complicated nor too simple) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| It was easy to find the information I was looking for                                         | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Examples were clear and easy to follow                                                        | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Illustrations added to my understanding of the software                                       | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| I liked the page design of the manual                                                         | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

If you have specific comments or if you have found specific inaccuracies, please report these on the back of this form or on a separate sheet of paper. In the case of inaccuracies, please list the relevant page number.

May we contact you further about how to improve SCO documentation?  
If so, please supply the following details:

Name \_\_\_\_\_ Position \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City & Post/Zip Code \_\_\_\_\_

Country \_\_\_\_\_

Telephone \_\_\_\_\_ Facsimile \_\_\_\_\_